Solar powered weather station

PV generator and mechanical design group

Morten Haas (4701852)
Aru van der Knaap (4693531)

TUDelft

Solar powered weather station

by

Morten Haas and Aru van der Knaap

for the
of the Delft University of Technology.

Abstract

This project deals with the design of a weather station that can work completely off-grid. Its only source of power must be an embedded PV panel. The energy generated by the PV panel can be directly used by the station itself or stored in a battery. Energy can be drawn from the battery when the energy from the PV panel is not sufficient or totally absent (for instance during night).
The design of this PV generator, tailored to the needs of the solar powered weather station, is vital to the effectiveness of the whole system. This thesis will discuss and cover, what measures should be taken to create a working prototype of a PV panel that can power the internal electronics of the weather station and what type of design is necessary to house the system as a whole. This design includes the housing of the electronics and the frame that is used to connect the panel with the housing. This frame could also be mounted to a pole. This also includes how many solar cells are necessary. Or the possible angles for the solar panel, and what influence they have on the lifetime and reliability of the system.

Preface

This thesis on the design of a PV generator and its framework for an Arduino as a weather station, would not have been possible without the help we received throughout the project. Firstly, we would like to give our thanks to our supervisor Patrizio Manganiello and proposer of the project for his guidance and support. Secondly, we would like to thank Mirco Muttillo for sharing his experience and helping us when needed during the lab sessions. Thirdly, we would like to thank Martin Schumacher and Ton Slats for their assistance in the Tellegen Hall. Finally we would like to give our thanks to our other group members, Tibbe van der Biezen, Henk van Grootheest, Maarten van der Meulen and Roan Föllings for their excellent cooperation during the project.

Contents

Nomenclature

Table 1: Nomenclature

Label	Description (unit)
PV	Photovoltaic
MPP	Maximum power point
STC	Standard test conditions
I_{sc}	Short circuit current (A)
$I_{sc,STC}$	Short circuit current at STC (A)
V_{OC}	Open circuit voltage (V)
$V_{OC,STC}$	Open circuit voltage at STC (V)
P_{max}	Maximum power point (W)
V_{mpp}	Maximum power point voltage (V)
I_{mpp}	Maximum power point current (A)
a	Diode ideality factor
N	Number of series connected cells
R_s	Series resistance (Ω)
$R_{s,STC}$	Series resistance at STC (Ω)
R_p	Shunt resistance (Ω)
$R_{p,STC}$	Shunt resistance at STC (Ω)
G	Solar irradiance (W/m^2)
G_{STC}	Solar irradiance at STC (W/m^2)
I_0	Dark saturation current (A)
I_{pv}	Photocurrent (A)
v_t	Thermal voltage (V)
q	Electron charge (C)
k	Boltzmann's constant (J/K)
T	Temperature (°C)
T_{STC}	Temperature at STC (°C)
K_v	Open circuit voltage temperature coefficient (%/K)
K_i	Short circuit current temperature coefficient (%/K)
K_p	Maximum power point temperature coefficient (%/K)

1 Introduction

1.1. System overview

The goal of this project is to make an autonomous weather station which only runs on solar energy. A large number of these stations are to be placed in and around urban areas where they will gather data about humidity, temperature, noise and particulate matter. This information will be uploaded via Wi-Fi to a database where it can be analysed. The results may be used in traffic planning, city-planning or to make some weather predictions. In order to make it feasible to have a lot of these stations, the stations themselves should be as small as possible, cost efficient and maintenance free for at least one year.

1.1.1. Structure

The project is divided into three groups, each focusing on a particular part of the station (summarised in figure 1.1).
The first group is responsible for the power generation of the Photovoltaic (PV) array. An additional task is the design of the housing and support structure of the station, as well as a mounting system to allow it to be attached to an object like a light pole.
The second group will focus on the part which connects the output of the PV-array to the micro-controller. The main subject is maximum power point tracking (MPPT), to get the maximum power transfer between the panel and the controller. This group is also responsible for the power management when it comes to battery charging and discharging.
The final group takes care of the choice and implementation of the sensors. Based on the power generation of the solar panels and consumption of the system, an algorithm is made to optimize the frequency of measurement. After collecting the data, they make sure it is periodically uploaded to a database via Wi-Fi.

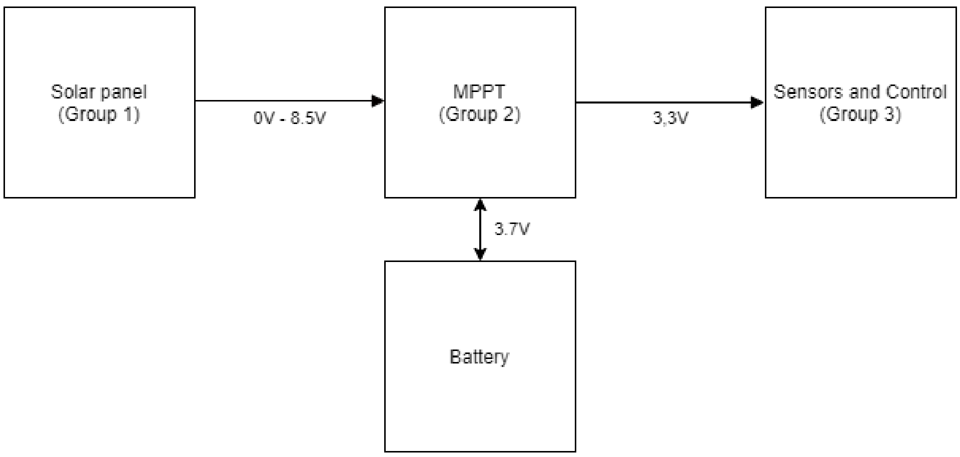


Figure 1.1: System overview with sub-group division.

2

Programme of requirements

2.1. Mandatory requirements

Functional requirements:

- The system must measure at an interval of at least one hour.
- The system must transmit the measured data at least once a week.
- The system must send its measured data over Wi-Fi.
- The system must measure:
  - Temperature
  - Humidity
  - Air pressure
  - PM2.5/PM5/PM10 pollution

Non-functional requirements:

- The system must function in an urban environment for at least 1 year.
- The only external power source must be PV.
- The maximum size of the box must be 250x250x300mm.

2.2. Trade-off requirements

- The system should contain a mounting system.
- It is preferable to also measure:
  - NOx
  - O_3
  - SO_2

3

Design requirements for the PV generation group

In addition to the selection of the main programme of requirements that are relevant for this group, other requirements were imposed on this group by the other two groups of this project to achieve the main goals.

3.1. Mandatory requirements

Functional requirements:

- The panel must have a V_{mpp} under standard test conditions of around 6V.

Non-functional requirements:

- The maximum size of the box must be 250x250x300mm.
- The only external power source must be PV.

3.2. Trade-off requirements

- The system should contain a mounting system.
- The panel should have a V_{mpp} above 5V under the worst practical conditions.
- The panel is preferred to have some kind of cooling technology.

4
State-of-the-art analysis

4.1. Working principle of Photovoltaic cells

The principle of a photovoltaic cell is the generation of potential difference at a junction between two materials [1]. The generation of charge carriers due to the absorption of photons in the materials that make up the junction, is where the difference in potential starts. These photons excite an electron to the conduction band from the valence band in a semiconductor material as shown in figure 4.1. The void left behind by the excited electron in the valence band is called a hole. Electrons and holes come in pairs and they will eventually recombine. If they do before being collected, Then their energy difference can not be used to generate usable current. However, by creating semipermeable membranes, which are usually made out of n- or p-type materials, depending on what type of charge carrier they allow through. N-type membranes allow holes through, while p-type membranes allow electrons through. These membranes allow movement in only one direction by either an electron or a hole, the electron-hole pair can be separated from each other and extracted from the solar cell with electrical contacts. Which allows them to perform work in an external circuit after which they finally recombine and the process can start over as shown in figure 4.2. The charge carriers are collected at the terminals of the junction, where the chemical energy is converted into electrical energy. Which can then be used to power external circuits or devices [1].

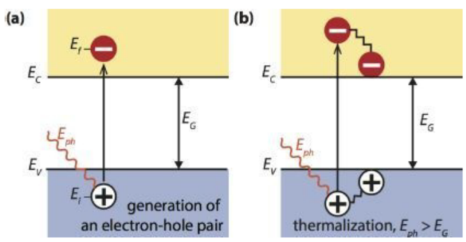


Figure 4.1: Generation of charge carriers with energy levels [1]

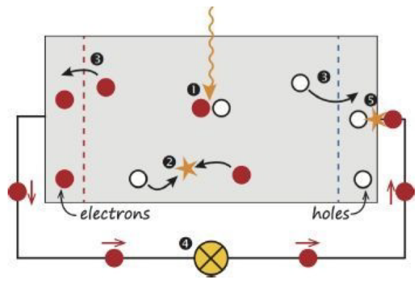


Figure 4.2: The working principle of a solar cell [1]

4.2. Cell Technology

The most common and researched PV cell technology is based on monocrystalline silicon, it is one of the most readily and commercially available PV cells on the market. However standard crystalline silicon solar cells usually come in 5 inch or 6 inch sizes. Another readily available PV cell on the market, is polycrystalline silicon solar cells. They are generally less efficient than monocrystalline silicon solar cells, although modern lab tested polycrystalline solar cells do compete with the efficiencies of monocrystalline silicon [2][3]. The commercially available smaller size solar cells are generally made out of polycrystalline silicon. Another way to obtain smaller size solar cells, is through cutting conventional c-Si cells. Other cell technologies were considered, like thin film or multi junction solar cells. Unfortunately thin film solar cells suffer from lower efficiencies, around 10% [4]. Efficiencies like 19%

for thin film are possible [5], however they are currently more a proof of concept. Rather than commercially available market options. While multi junction solar cells do reach high efficiencies, higher even than conventional monocrystalline silicon, they are excessively expensive as they are not yet readily available for the consumer market. [6].

4.3. Solar modules

The interconnection between cells to form strings is also a vital part of PV technology design. Most commonly a single module is a string of series connected solar cells, because the series connection results in the module having a higher voltage for the application. Parallel connection can also be done, but because parallel connection results in higher currents, and therefore higher resistive losses, this is generally avoided [1]. PV modules commonly have bypass diodes integrated into them, these are necessary for the PV modules to function optimally. For a multitude of reasons, a cell in a series connected string of a module could stop functioning properly, such as shading or damage to interconnects. This would cause the current of the entire module to be limited, or even cause the module to stop functioning all together. In order to circumvent this, a bypass diode is installed alongside the solar cells. Usually a string of a few solar cells has a single bypass diode [7]. This mitigates the effect of current limitations, but to ensure that a single cell can not have such a great negative effect on the performance of the module, in some cases each and every cell is connected to bypass diodes [8]. Connecting every single solar cell in a module with a bypass diode is more costly, but some solar cells come with build in bypass diodes. Which makes interconnection easier and also cheaper than connecting solar cells with external bypass diodes, however this is currently not yet available on the market, despite research on it [9].

4.4. Position of the sun and module orientation

Since the power production of a solar cell relies on the sun, the position of sun is crucial to the magnitude of the generation. The position of the sun can be defined using two components: the azimuth and the altitude. The azimuth is the angle on the horizontal plane from north for the observer in a clockwise direction, which is from 0°to 360°. The altitude is the angular elevation from the center of the disk in the horizontal plane, ranging from -90°to 90°, as depicted in figure 4.3. Another crucial element, while determining the efficiency of a PV module, is the orientation of the module. The angle perpendicular to the surface of the module can also be expressed in terms of azimuth and the altitude, where the altitude is now reliant on the angle of the surface of the module and the ground ($altitude = 90° - \theta_m$), as shown in figure 4.4. In order to maximise the generation, these two positions have to be matched. Sunlight arrives through the atmosphere, partially scattered, in two major components: the direct beam component and the diffuse light. The magnitude of both components is important to determine the effectiveness of a PV system, since they determine how much power the system could generate.

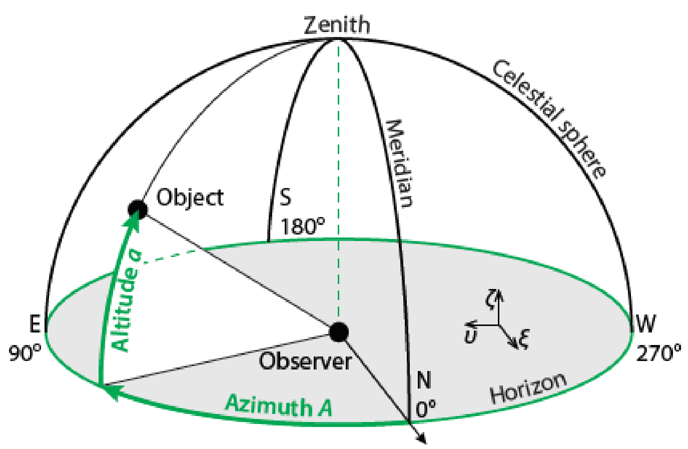


Figure 4.3: Illustration of the azimuth and altitude in the horizontal coordinate system [1].

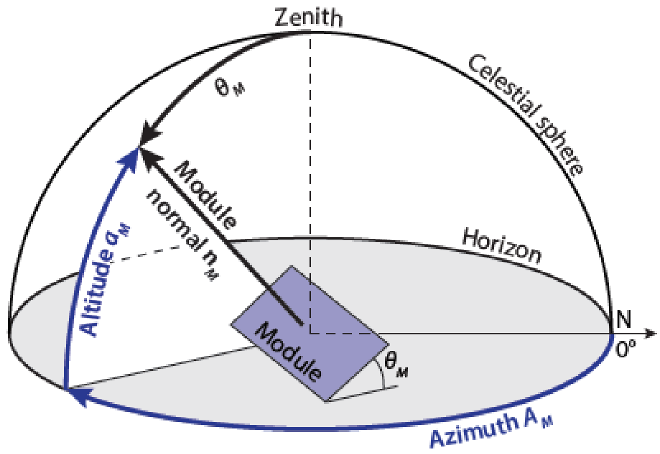


Figure 4.4: Illustration to determine the orientation of a PV module with angles[1].

4.5. Shading

Shading or partial shading of PV arrays is a big issue for PV generators, as their power generation could be significantly limited. When complete shading happens there is no way to limit the reduction in generation. However, the effect of partial shading can be limited through proper use of PV array configurations.
If a cell in a series array is shaded, then the current generation of the whole array is bound by this lower current of the shaded cell. This could greatly reduce the power generation of the whole array instead of just this cell. To combat the effect of partial shading of a single module, previously mentioned bypass diodes are used. Pv arrays also use bypass diodes to ensure the best performance. However, bypass diodes can reduce the overall system performance by introducing multiple peaks. Alternative methods of mitigating the effect of partial shading include PV array configurations, these methods also suffer from multiple peaks [10]. So a case can be made for both methods.

A parallel array configuration, where every module is connected in parallel, is innately the least limited by partial shading. As the current of one module can never affect the current of another, but this results in a very low voltage and very high current for the array, which is not very desirable for practical applications. Instead a Total cross-tied (TCT) or Honey Comb (HC) array configuration might be used. Where TCT (shown in figure 4.5) is the most efficient with symmetrical PV array configurations and HC (shown in figure 4.6) is the most efficient with asymmetrical PV array configurations [10][11]. A symmetrical PV array configuration would be a configuration, where the number of modules connected in series, is the same as the number of modules connected in parallel, ie. a 3 by 3 array. An asymmetrical array configuration, is a configuration where the number of modules connected in series is not the same as the number of modules connected in parallel, ie. a 3 by 4 array. An example of an asymmetrical array is shown in figure 4.5 or figure 4.6

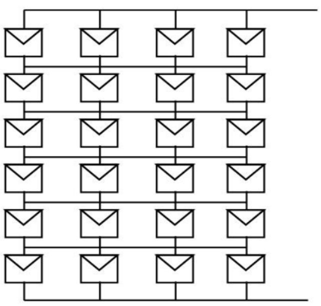


Figure 4.5: TCT array [10]

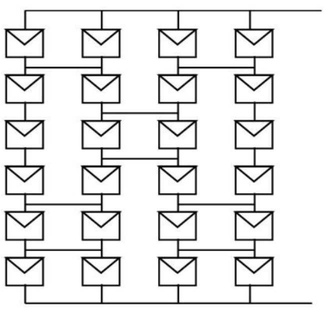


Figure 4.6: HC array [10]

4.6. Half-Cell Technology

Full-size photovoltaic cells are the most commonly used design in commercial solar modules. A new design approach is by using half-cut cells to reduce cell-to-module losses [12]. This is mostly caused by the current being approximately halved as can be seen in figure 4.7, thus reducing resistive losses. This results in lower average heat dissipation [13]. Also light collection appears to be improved by the increased cell gaps in half-cell modules [14]. Half-cells also tend to be better resistant to the effects of partial shading. When one half of the cell is shaded, the other half does not have to be and its bypass diode would therefore not need to be active thus reducing the shading losses by up to 50% [15].
Half-cells do introduce a couple of other problems that need to be dealt with: An extra step in the production line for laser cutting, edge recombination issues from the laser cutting of the cells, losses from mismatch of the cut cells, more space is required due to additional spacing from the half-cells, cut cells also tend to be more fragile than its full-cell counterpart [16] [17]. Despite these added difficulties, and thereby added price. Half-cell modules are becoming more popular compared to conventional PV modules [18].

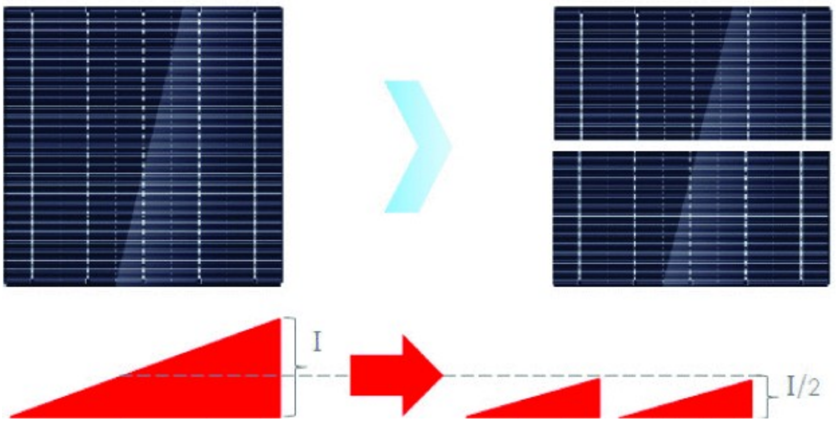


Figure 4.7: Full-cell and half-cell comparison [16]

4.7. Thermal management

The performance of the solar cells of a PV module are heavily dependent on their temperature. The operating voltage of a module, decreases as temperature increases, lowering overall power generation. Furthermore, the temperature of the solar cell is dependent on the ambient temperature in conjunction with the solar irradiance, the module is subject to [1]. This results in the temperature of the module being significantly higher than the ambient temperature, further reducing performance. Therefore a way of reducing or managing the temperature of a module should be accounted for in a proper design of a PV module.

5

Design Process

5.1. Design overview

This chapter discusses the steps taken to design the PV and physical design of the weather station. First the choices regarding the solar cell will be discussed. The following sections go into the design process of the other factors of the weather station.

5.2. Solar Cell

5.2.1. Necessary generation

The choice of solar cell and the amount of solar cells necessary to realise the prototype, heavily depends on the actual solar irradiance and the consumption of our system. In order to estimate the amount of generation necessary, a methodology that does not depend on the individual characteristics of available solar cells had to be created. Therefore, the Matlab [19] script that was designed is only depended on the actual weather data of the test location, in this case the city of Delft.

The implementation is simple: calculate the level of charge of the battery throughout the testing period, by continuously adding the average generation each hour and also subtracting the average consumption each hour. The generation is calculated by multiplying the solar irradiance of our test site by the area of our panel and an efficiency factor. In this basic design, the efficiency is the total efficiency of the PV generator, including the MPPT. The consumption is taken as the average consumption in an hour in this preliminary version. This gave us an estimated constant consumption of 0.1 W, which was found from some preliminary studies on the sensors by the other groups. Ideally both the efficiency factor and the consumption are calculated in a more nuanced way, however for this preliminary estimation, this is sufficient. The results found in figure 5.1 show that according to this model, the system can survive the whole testing period of 12 years with a minimum of two batteries with an effective capacity of 7.2 Wh each, irrespective of the parameters of the actual solar cell, as long as a minimum area and system efficiency of 0.0625 m^2 and 10% respectively are reached. As can be seen in figure 5.1, the system does not drop below 40% capacity, thus it was decided to be a good starting point for the design with an acceptable margin of error.

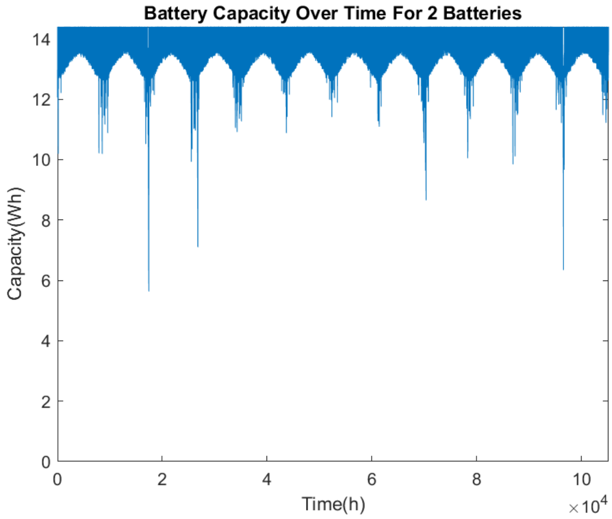


Figure 5.1: The level of charge in the battery over a 12 year period

5.2.2. Choice of solar cell

Market Options

Three different solar cell models were considered during this analysis: The Velleman SOL3N, AKW ETS6-2040 and a mystery solar cell. The focus of this analysis lies on finding a good solar cell for a small weather station. Only these three solar cells were considered, because others could not be delivered within a reasonable time.

Velleman SOL3N

The Velleman SOL3N [20] is a small polycrystalline solar cell. With a total area of 0.0018 m^2, it is the smallest of the three solar cells, as can be seen in table 5.1. However, its conversion efficiency is the lowest of the cells, but it partly makes up for that with its much lower current thus decreasing resistive losses.

AKW ETS6-2040

The AKW ETS6-2040 [21] is a 6 inch monocrystalline solar cell. It has the highest efficiency, as well as the highest maximum power of the three cells. One downside of this cell is that its current is also very high, which increases the resistive loses, and it has a high surface area. Although this cell could be cut to reach a higher voltage, it is held back by its low voltage compared to its large surface area. This means that the area of the PV panel would increase substantially when using this solar cell. If the same V_{mpp} must be reached.

Mystery solar cell

The last cell is of unknown origin [22]. And its product name is not known to us, therefore 'mystery' solar cell. Its size is in between the other cells with a slightly lower efficiency than the AKW ETS6-2040, but it does have a much lower current. This is also the most efficient of the monocrystalline based cells to cut when looking at the effective area, due to it having almost the same voltage and spanning a significantly smaller surface area.

Table 5.1: The Three different solar cells compared with each other[20] [21] [22]

Model	Velleman SOl3n	AKW ETS6-2040	Mystery solar
Type	Polycrystalline	Monocrystalline	Monocrystalline
Area (mm)	46 x 40	156.75 x 156.75	125 x 125
Efficiency (%)	10.90	20.40	17.86
Pmpp (W)	0.2	4.95	2.79
Umpp (V)	1.0	0.547	0.527
Impp (A)	0.2	9.05	5.292
Uoc (V)	1.2	0.654	0.628
Isc (A)	0.22	9.61	5.669
FF (%)	75.76	78.76	78.34

Comparison of the cells

While deciding between the cells, the following factors where considered the most important: The area, the efficiency and the voltage of the cell. Since the space for the generation is limited, only one of the AKW ETS6-2040 solar cells could fit on our panel, which would not produce a high enough voltage and power to meet the requirements despite having the highest efficiency. While it could be cut to allow for less total cells, a lot of cuts would need to be made for it to reach a sufficiently high voltage. Although the Velleman SOl3n is quite small compared to the other cells and it has the highest voltage of the cells, its efficiency is also much lower than the other cells due to its polycrystalline nature. This would mean that almost twice the amount of area needs to be covered in order to compare to the generation of the other cells. Ultimately, the mystery solar cell was considered the optimal choice since it has a reasonable efficiency compared to the others and it could be cut into smaller cells. By cutting this cell and connecting the cut cells in series, the needed voltage could be achieved, this could also lower the resistive losses of the final PV module. Since it has about the same voltage as the AKW ETS6-2040 solar cell, but a much smaller size, it would take a much smaller area to reach the needed voltage.

5.2.3. Cutting of the cell

As stated in section 4.6, cutting has an adverse effect on the fragility of the cell. Therefore, the minimum amount of cuts needed to achieve the desired maximum power point voltage were made. As can be seen in chapter 3, this is about 6V. This can be reached by cutting 4 different cells, 3 times each for a total V_{mpp} of 6.324V. This allows for a bit of leeway in the voltage from imperfections of the cells.

5.2.4. Modeling of the cell

Multiple models exist to describe the operation of photovoltaic cells under illumination. The most commonly used representations are the ideal single diode model, the single diode Rs-model, the complete single diode model and the two-diode model [23].
The ideal single diode model is generally not used for simulation, shown in figure 5.2, since it does not account for the resistive losses within the cell. The single diode Rs-model, shown in figure 5.3, tries to represent this by including a series resistance. However, the accuracy of this model decreases when the temperature variation is too large, because it does not account for the effect of K_v on the V_{OC} [23]. This is improved by the complete single diode model, shown in figure 5.4, with the inclusion of a shunt resistance. This keeps the model relatively simple, while also making it quite precise. The most complicated of these four models is the two-diode model, shown in figure 5.5. This model tries to include the thus far neglected recombination current loss in the depletion region by adding another diode. The mathematical model of each of these equivalent circuits are represented in equation 5.1-5.4 [24].

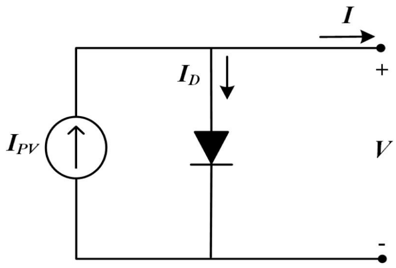


Figure 5.2: The ideal single diode model [23]

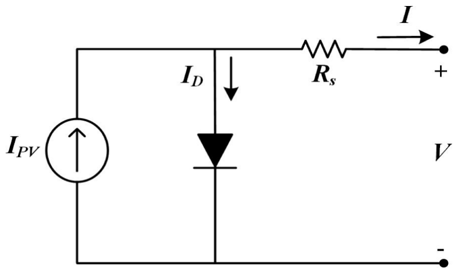


Figure 5.3: the complete single diode model [23]

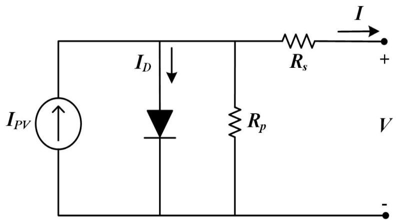


Figure 5.4: The single diode Rp-model [23]

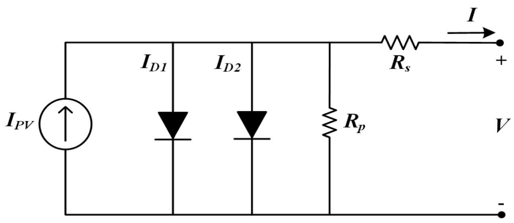


Figure 5.5: the two-diode model [23]

$$I = I_{PV} - I_0 \left[exp\left(\frac{qV}{akT} \right) - 1 \right] \tag{5.1}$$

$$I = I_{PV} - I_0 \left[exp\left(\frac{q(V + IR_S)}{akT} \right) - 1 \right] \tag{5.2}$$

$$I = I_{PV} - I_0 \left[exp\left(\frac{qV}{akT} \right) - 1 \right] - \frac{V + IR_S}{R_p} \tag{5.3}$$

$$I = I_{PV} - I_{01} \left[exp\left(\frac{qV}{a_1 kT} \right) - 1 \right] - I_{02} \left[exp\left(\frac{qV}{a_2 kT} \right) - 1 \right] - \frac{V + IR_S}{R_p} \tag{5.4}$$

Ultimately, the single diode model was chosen to be suitable for the purpose of this project due to its simpler design compared to the two-diode model without sacrificing too much precision [23]. Since it only has one exponential, it is much faster to simulate. Furthermore, it is easier to extract from a measured I-V curve the several parameters needed to use this model for simulation.
The five parameters that need to be determined are a, I_0, I_{pv}, R_s and R_p. An iterative algorithm was implemented in Matlab [19] based on the method presented in this paper [25]. The code, that can be seen in appendix B.1.2, is split into two main steps. Firstly, it estimates the a and R_s, which it then proceeds to use to calculate the other three values.
Once these parameters were determined, they were used to construct an LTspice [26] model of the single diode model. The circuit can be seen in figure 5.6 and the corresponding code for the model is in appendix B.2.1.

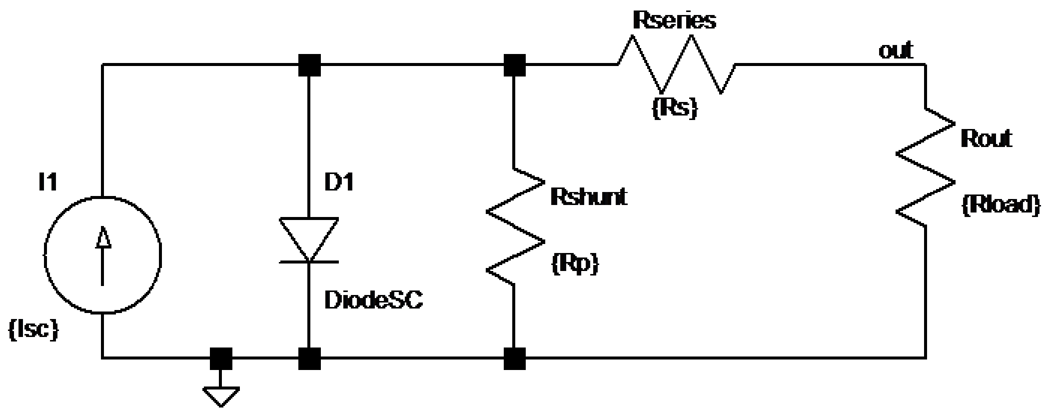


Figure 5.6: Single diode R_p-model

Several equations were used to assist in the creation of this model [25]. Equation 5.5 shows the linear effect of the irradiance on the shunt resistance.

$$R_p = R_{p,STC}\frac{G_{STC}}{G} \tag{5.5}$$

From equation 5.6 the relation between the I_{SC} and the irradiance and temperature can be seen. This shows that irradiance has a larger effect on the current, than the temperature, since K_i is quite small compared to the $I_{SC,STC}$.

$$I_{SC} = I_{SC,STC}\frac{G}{G_{STC}} + K_i(T - T_{STC}) \tag{5.6}$$

Equation 5.7 shows that, unlike the I_{SC}, the V_{OC} has a slightly stronger relation with the temperature and a lesser one with the irradiance. This is due to K_v generally being larger than K_i and V_{oc} only having a logarithmic relationship with the G.

$$V_{OC} = V_{OC,STC} + K_v(T - T_{STC}) + v_t ln\left(\frac{G}{G_{STC}}\right) \tag{5.7}$$

From equation 5.8 the new a can be calculated by rewriting the equation, as can be seen it is dependent on the V_{OC} and the I_{SC} [27]. These change in different degrees when the irradiance and the temperature fluctuate, thus possibly effecting a.

$$V_{OC} = \frac{kaT}{q} ln\left(\frac{I_{SC}}{I_0} + 1\right) \tag{5.8}$$

5.3. Panel Temperature

The cell temperature of the PV module is generally not the same as the ambient temperature, as the module generates and stores heat. Due the irradiance it is subject to or the module heating up because of the current running through it. This cell temperature needs to be estimated, because it directly impacts the performance of the PV module. Two of the most commonly used prediction models are the nominal operating cell temperature (NOCT) model and the Sandia National Laboratory (SNL) model [28].

The NOCT model uses a steady state approach for the prediction of the temperature, thus assumes temperature to be time independent, within a period of 1 hour. To find the NOCT, an experiment needs to be done under specific conditions. The conditions are a G of 800 W/m^2, an ambient temperature of

20°C and a wind speed of 1 m/s. This value can then be used with formula 5.9, where T_m is the module temperature and T_{amb} the ambient temperature, to estimate the module temperature. One limitation of the model is that it can produce significant errors when the installation conditions are not the same as the standard conditions [28].

$$T_m = T_{amb} + \frac{NOCT - 20}{800} * G \tag{5.9}$$

The SNL model seeks to improve the flaws of the NOCT model. It does this by taking into account the wind speed (V_w), as well as the encapsulation of the PV and the type of installation. As shown in equation 5.10 with a and b the two empirical coefficients, this model is more versatile than the NOCT model. However, it is still based on empirical conditions and can therefore vastly vary based on the site conditions and setup.

$$T_m = T_{amb} + G * exp(a + b * V_w) \tag{5.10}$$

Both models tend to overestimate the T_m, as they do not account for the effects of the heat losses due to the thermal radiation process [28]. This can cause a mean bias error (MBE) up to 21% of the mean temperature for the NOCT model and approximately half of that for the SNL model [28]. Although this temperature difference can be quite high, this generally only leads to a difference in the expected power of about 6% [28]. Since the difference between the expected power is not that high and only the NOCT needs to be estimated, this model was chosen to determine the T_m. Although the error in the T_m estimation is lower for the SNL model, it is also more complicated to derive the parameters as they are site and system dependent. For these reasons the NOCT model was chosen over the SNL model, as it was deemed good enough for the purpose of this project.

5.4. Thermal Management

There are a couple ways to improve the cooling of the panel. Only passive cooling options have been considered, because others would need additional power. This could have an adverse effect on the energy efficiency of the system, as well as increasing the complexity of the design [29]. The first considered option is to simply keep the backside of the panel open and let it naturally dissipate heat in this way. This is the simplest solution, but also the least effective one as air does not have a very high heat capacity and no additional measures are taken to increase heat dissipation.

One of the options considered is to add fins to the backside of the panel. These fins would increase the surface area of the backside and allow for more air flow. This could result in an average decrease in temperature of up to 7°C compared to the open backside, as shown in this paper [30].

Another option is to use the fact that temperature difference creates a force that can push a fluid through a pipe and transfer heat in this way. These vacuum sealed heat pipes transport this heat to another place, where it can cool down, and then returns to the panel to absorb more heat and repeat this process [31]. This can result in an average decrease in temperature of about 10°C [32] in comparison to the first option.

From these three options, the first solution was chosen for the following reasons. While the other considered achieve lower temperatures and thus higher efficiencies, the difference in performance would not be significant enough for this project. As the amount of generation is not large enough for it to significantly matter for this setup. Although on could argue that as much as energy as possible should be saved because this is a self-powered device, it can be quite difficult to create custom sized fins in the right dimensions. Therefore, the inclusion of fins in this design were avoided. In the case of the heat pipe system, it would significantly increase the complexity of the system and the likelihood of failure for the system due to possible leakage of the pipes or other issues.

5.5. The Housing

The design for the housing of the electronics needs to take into account a couple important factors. The most important one is the maximum size of the box of 250x250x300mm, which is based on the limitations of the 3d-printer available at TU Delft laboratories. This is the reason for the limitation on the size presented in the programme of requirements. Another key element is the different needs of the sensors that have to be taken care of. The most important of those needs is that some sensors require a direct flow of open air in order to function optimally, while others prefer to be enclosed. The sensors that need this outside contact are the noise, PM and temperature/humidity sensors [33] [34] [35]. The datasheets for these three sensors can also be found in appendix C. The PM sensor should also not be too close to the noise sensor as its fans can produce some noise. The PM sensor also has both input and output openings for the air that need to be slightly separated to minimise the testing of the same air. The pressure sensor and the other electronics are the components that need to be enclosed [36].

In order to accommodate for the needs of both the sensors that need outside contact and the components that need to be enclosed, it was decided to design two separate compartments according to their needs. Since the PM sensor and the noise sensor should not be right next to each other, the maximum distance between them within the same compartment can be achieved by placing them on opposite sides of the compartment. Since contact between the two compartments is necessary to interconnect the sensors and the PV panel with the control and power electronics, it was also decided that one hole between the compartments and the top of the housing was needed for wiring. These holes allow for air to flow between the compartments and the top of the housing, to minimise this effect it should be partly closed to decrease the space between the wires and housing. The considered options to decrease this space are a custom lid with holes for each wire that can be attached to the housing with a rubber ring or to close it using tape. The tape option is the easiest to implement, but is also the least reliable as it is prone to wear and tear. Therefore the custom lid was deemed more appropriate even though it would be more complex to design and implement. Since some moisture is guaranteed to form in the compartment with open air contact, some way to release it is also needed. The easiest way, which was also chosen, is to add a small slope to the bottom of the compartment with a small hole at the end for the moisture to escape from.

5.6. The Mounting Frame

In order to connect the panel and the box with each other, a separate frame is necessary. This frame can then also be used to mount the weather station and help support the weight of the panel, because the box would probably not be able to support the full weight. Since this needs to be able to withstand the weight of the whole panel as well as the housing, a strong yet lightweight material was needed. The two considered options are aluminum and stainless steel. Aluminum is the lightest of the two materials, but also slightly weaker. Stainless steel is stronger at the cost of a much higher weight. Since the weight of the rest of the system is not that high, the panel itself weighs up to 2 kilos and the housing with electronics weigh less than 1 kilo. Aluminum was chosen as it should suffice to carry the system and is also the lightest of the two. It will keep the panel at a fixed angle, instead of allowing for adjustments to lower the amount of moving parts that could break. It will also hold the box mainly from the back and the bottom to make sure that the weight on the box is minimal. In order to mount the complete the design a hose clamp would be used to connect the system to a pole, as this is also regularly used to connect signs and other objects to poles.

The idea for the frame has been sketched, as shown in figure 5.7 and 5.8. Figure 5.7 shows how the PV panel and the housing rests on the frame and the backside of the panel is kept open. The system could be mounted on the upper backside of the frame with a hose clamp.

Table 5.2: Things that are represented by the colours in figures 5.7 and 5.8

Colour	Name
Blue	PV panel
Black	Frame
Red	Housing

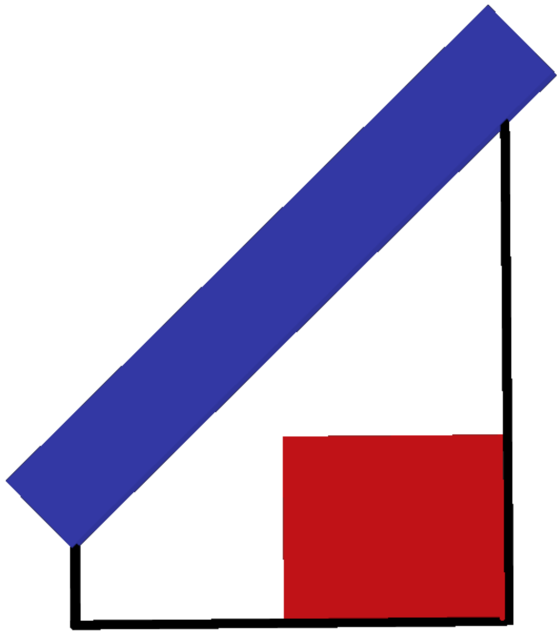


Figure 5.7: Frame sketch side view

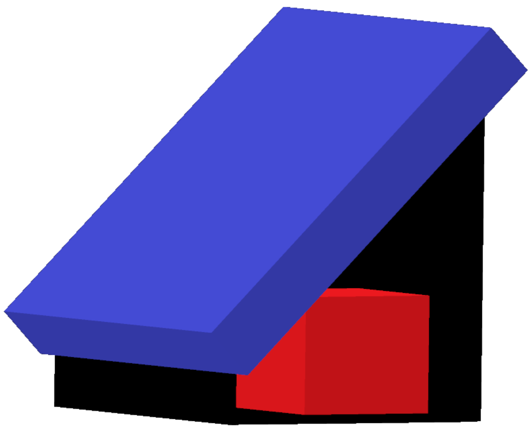


Figure 5.8: Frame sketch neutral view

6

Prototype implementation and validation results

6.1. Prototype overview

This chapter is dedicated to the final design of the prototype. How this design came to be, what types of testing and analysis were done, to prove the validity of the final design, are covered here along with their results.

6.2. Basic PV simulation

In order to simulate and test a digital version of our PV panel, we created two different models. One in LTspice, and one in Matlab. The Matlab model allows for complete control over the characteristics of the diode; the LTspice model instead uses a diode as a black box model, where we are unable to figure out the exact characteristics of the diode. The same equations were used to build either model. The parameters required by these equations to build the model, stem from the physical prototype. The prototype was tested and measured using a solar simulator for standard test conditions. The results of this test are shown in figure 6.1. From these measurements the five parameters for the diode model are extracted using the method described in section 5.2.4.

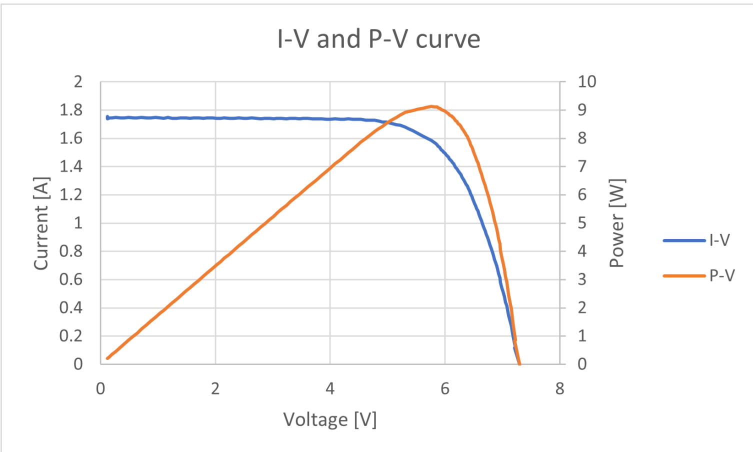


Figure 6.1: IV and PV curve of prototype

The results from the Matlab and LTspice models based on the created PV panel, are shown in the figures 6.2, 6.3, 6.4, 6.5.
Upon analysis of these figures, the main difference between the two models is the V_{oc}. It seems that our LTspice model is more sensitive to temperature, and that the V_{oc} especially is more sensitive to

temperature. It is worth to note, that this causes a difference in maximum power point between the two models. It is impossible to say which version is more correct than the other, and more than likely both version are at least a bit wrong. This is further corroborated by calculating the average K_p of both models and comparing them to the known K_p of the panel from the data sheet.

Table 6.1: Comparison between 3 K_p

Variable	Matlab	LTSpice	Data sheet
K_p	-0.4198 (%/K)	-0.4805 (%/K)	-0.46 (%/K)

According to table 6.1, both models show a temperature coefficient that is different from the one mentioned in the datasheet. The LTSpice model is slightly higher and would result in a slight overestimation of the effect on the performance by the temperature. On the other hand, the Matlab model would underestimate the temperature effect on the performance. It is, however, important to note, that the calculation results only in an estimation, so it is still difficult to be sure, without much more testing on the physical panel itself.

Returning back to the figures 6.2 to 6.5. The temperature dependency could be an issue for the performance of our weather station. So much so, that with the current panel, the trade-off requirement for the V_{mpp} of 5 volts under worst practical conditions is not reached as shown in figure 6.2, if the module temperature is at 65 degrees for both models. On the other hand, our mandatory requirement of a V_{mpp} around 6 volts under standard test conditions is reached, as shown in table 6.2.

Table 6.2: Minimum V_{mpp} from figures 6.2 to 6.5

	5 °C	25 °C	45 °C	65 °C
100 W/m^2	6.6 V	6.13 V	5.49 V	4.81 V
200 W/m^2	6.6 V	6.10 V	5.49 V	4.86 V
300 W/m^2	6.6 V	6.09 V	5.46 V	4.85 V
400 W/m^2	6.6 V	6.06 V	5.39 V	4.78 V
500 W/m^2	6.6 V	6.03 V	5.42 V	4.77 V
600 W/m^2	6.5 V	6.00 V	5.35 V	4.77 V
700 W/m^2	6.5 V	5.95 V	5.31 V	4.70 V
800 W/m^2	6.5 V	5.94 V	5.32 V	4.68 V
900 W/m^2	6.4 V	5.88 V	5.27 V	4.66 V
1000 W/m^2	6.4 V	5.87 V	5.24 V	4.65 V

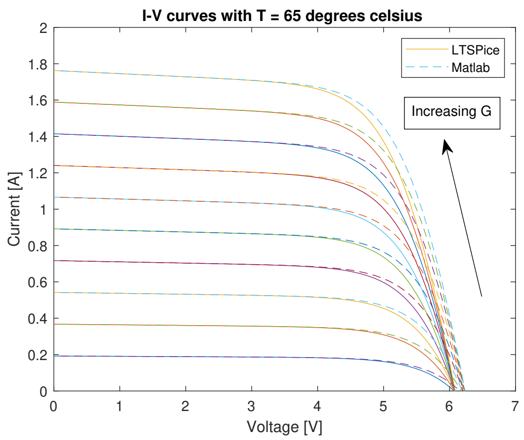


Figure 6.2: IV curve with increasing irradiance, from 100-1000

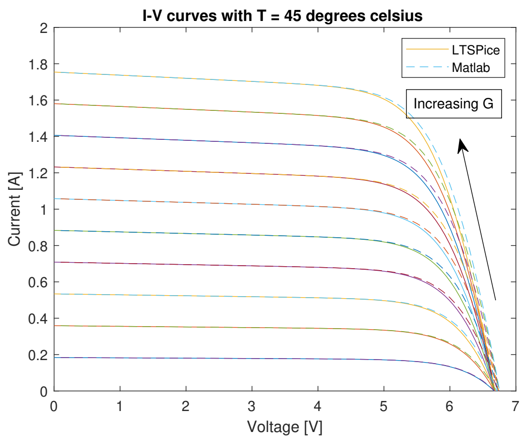


Figure 6.3: IV curve with increasing irradiance, from 100-1000

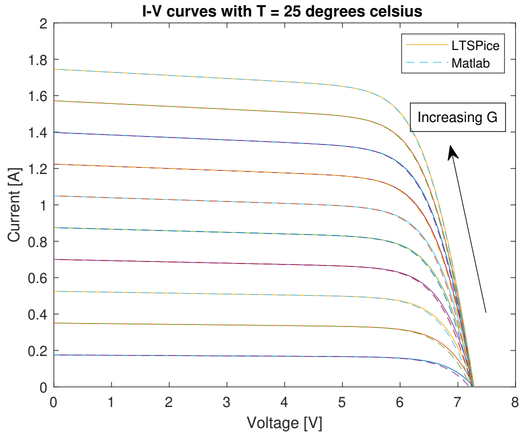


Figure 6.4: IV curve with increasing irradiance, from 100-1000

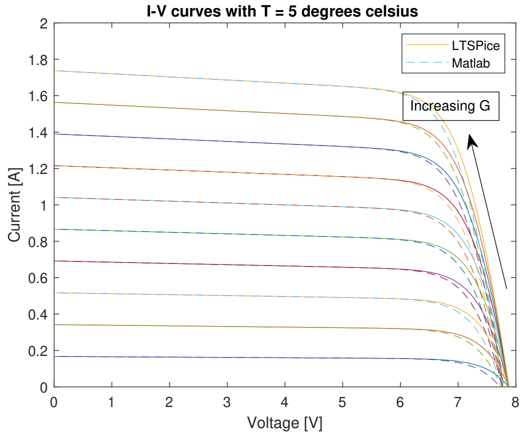


Figure 6.5: IV curve with increasing irradiance, from 100-1000

6.3. Advanced PV simulation

Between the LTspice model and the Matlab model of our PV panel, the Matlab model is more useful, even if the temperature dependency is slightly less accurate. The Matlab model is more convenient, as it can be converted into a function. Which makes it easier to use it to digitally test our PV panel with real world weather data, from our testing site.

We have considered three different angles for the PV panel: 76 degrees, 38 degrees and 29 degrees. 76 degrees is optimal for December, as the sun reaches its zenith at 14 degrees in December [37] (from here on referred to as winter angle). 29 degrees is optimal for June, as the sun reaches its zenith at 61 degrees in June [37] (from here on referred to as summer angle) and 38 degrees is the average optimum angle for a year [37] (from here on referred to as optimal angle). The software from [37] provides us with the time-series for plane-of-array irradiance at the different module tilt angles, together with the time-series of ambient temperature. Both with hourly resolution. With this data we could then estimate how much power is generated by the PV panel throughout the year, using the Matlab model. The full data set contains 12 years of solar irradiance and ambient temperature data, with hourly resolution. While extensive and useful for testing purposes, this is too much data to use as input for the Matlab model. The simulation time would take days to run. So due to time constraints, an alternative was used. A look up table (LUT) was created for specific ambient temperatures and solar irradiance. The LUT has a solar irradiance range of 50 W/m^2 to 1200 W/m^2, increasing with steps of 50 W/m^2. The ambient temperature range consisted of 5°C to 45°C, with steps of 5°C. For the cell temperature, we use the NOCT model described in section 5.3. The NOCT parameter was calculated using a physical test that was done. The ambient temperature, solar irradiance and cell temperature was known during this test. The results of this test and the resulting NOCT are shown in table 6.3. For each possible combination of solar irradiance and ambient temperature, the model was run and a P_{max} calculated. Although this LUT could also have been created from the LTSpice model, it is easier to create and change this table in Matlab.

Table 6.3

ambient temperature	cell temperature	solar irradiance	NOCT
20 °C	46.2 °C	1000 W/m^2	42

The data sets corresponding to each of the different angles were then divided by the step size, for both solar irradiance and ambient temperature. For a best case scenario, the solar irradiance is rounded up and the ambient temperature is rounded down. While for a worst case the solar irradiance is rounded down and the temperature is rounded up. To make analysis easier only one year of data is plot. The year with the least amount of generation in the winter was chosen to analyse. Furthermore to make the figures more clear, the hourly data for each day was summed.

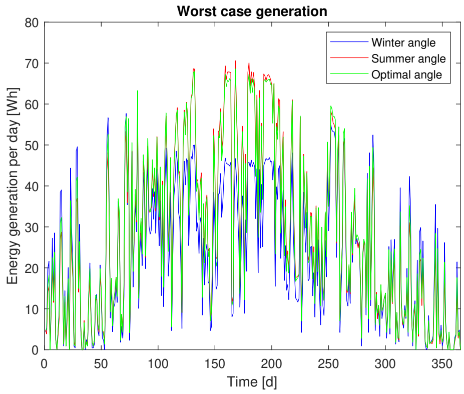


Figure 6.6: The total generation per day for a year

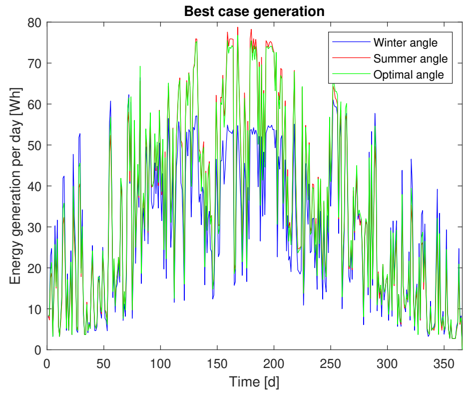


Figure 6.7: The total generation per day for a year

Figures 6.6 and 6.7 show the results. For further clarity the energy production in a year was added, as an average of 12 years, in the appendix, figures A.1 and A.2. Figures 6.6 and 6.7 show that the angle optimised for winter generates more power than the other angles in winter, while the same is true for the summer angle and summer. Which is what was expected. However, interesting to note is that, while the winter angle does very poorly in summer compared to the other two angles and the summer angle also does very poorly during winter, The optimal angle does remarkably well all year round. Only in the heart of winter or the heart of summer is the optimal angle outdone in energy generation. The difference between the summer and optimal angles, especially, is not very noticeable at all. Suggesting that the performance of the weather station is not much different with either angle, while the winter angle does seem to fall short. The difference between figures 6.6 and 6.7, for each angle, is quite large during the summer months. There is a significant difference during winter months, but not nearly of the same magnitude.

In order to further test the different angles for the PV panel, an updated version was used of the Matlab script used in section 5.2.1. This version uses a more accurate representation of the consumption with an optimisation algorithm, devised by subgroup 3: Sensors and Control. This allows for greater insight into the battery charge of the weather station, with each angle for the PV panel. The results found in figures 6.8 to 6.13 are quite surprising. The difference between best and worst case for each angle is much larger than the difference between each angle. Furthermore, with this model all angles are sufficient, though the winter and summer angle are a little bit less reliable in the worst case scenario, as it reaches a lower charge. It also is not the case that the winter angle increases the amount of effective generation during the winter significantly. The battery also seems to be full or close to full most of the time, especially in the summer, regardless of the angle. This indicates that there might be too much generation. For these reasons using the optimal angle seems to be the most appropriate. As neither the additional generation during the summer nor the winter for the other two angles is significant enough to consider them over the optimal angle.

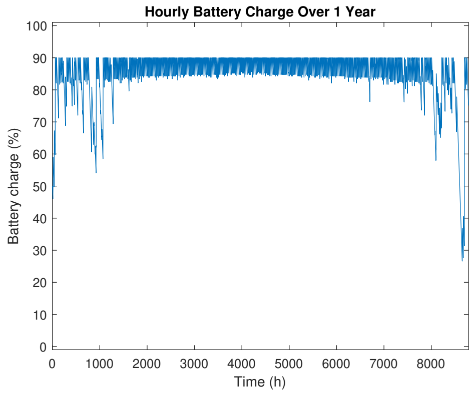


Figure 6.8: Worst case battery charge with winter angle

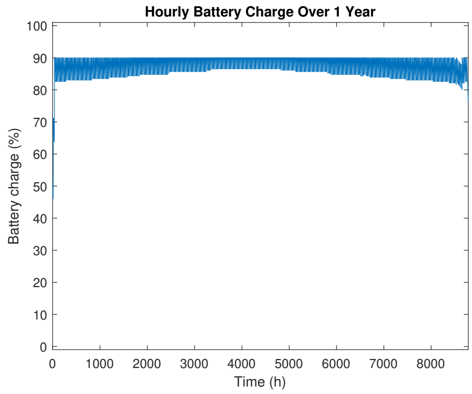


Figure 6.9: Best case battery charge with winter angle

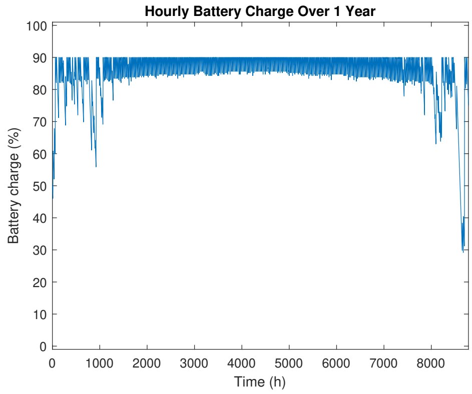


Figure 6.10: Worst case battery charge with optimal angle

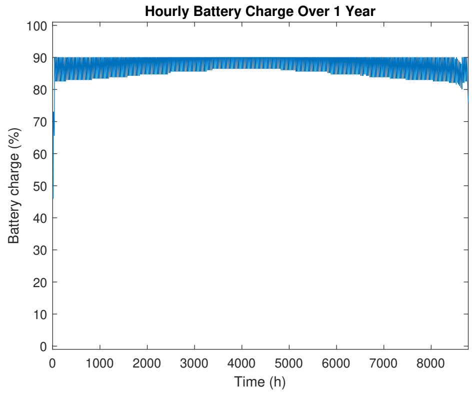


Figure 6.11: Best case battery charge with optimal angle

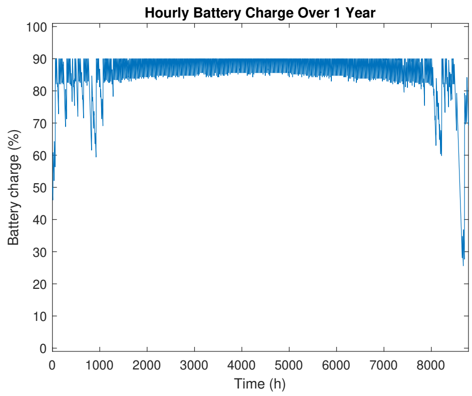


Figure 6.12: Worst case battery charge with summer angle

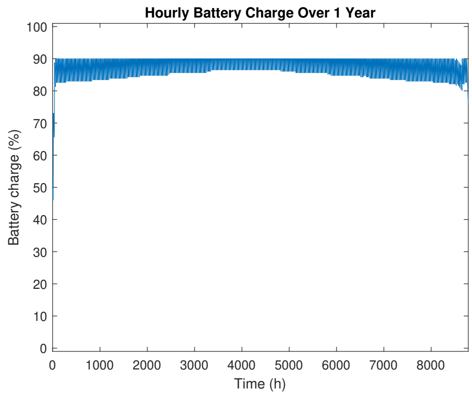


Figure 6.13: Best case battery charge with summer angle

6.4. Thermal Management

The final design does not have anything to improve the thermal management aside from an open backside, as stated in section 5.4. The average temperature of the module during the day from 7:00-18:00, as shown in figures 6.14-6.15, is not that high during the colder half of the year, but it hovers around 20-40°C during the other half of the year. As shown in figures 6.16-6.17, the maximum temperature each day of the module does not go past 45°C aside from a couple days each year in the worst case estimation. This means that it does not really matter that the system might not reach the trade-off requirement V_{mpp} of 5V around 65°C, as the temperature of the module should not come close enough to that value in practice. Therefore, the system stays above the required V_{mpp} of 5V for the worst practical conditions in this data set.

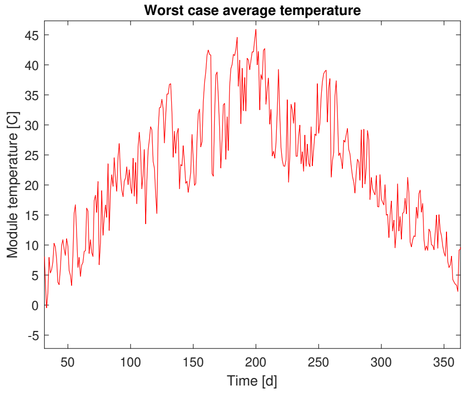


Figure 6.14: Worst case average temperature

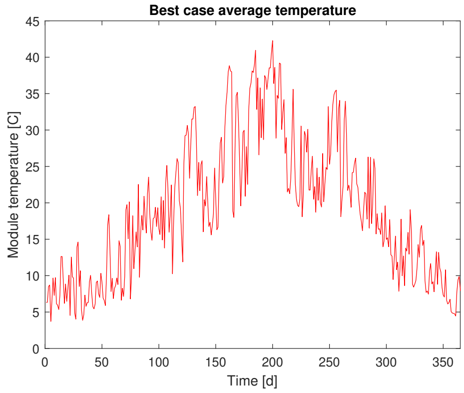


Figure 6.15: Best case average temperature

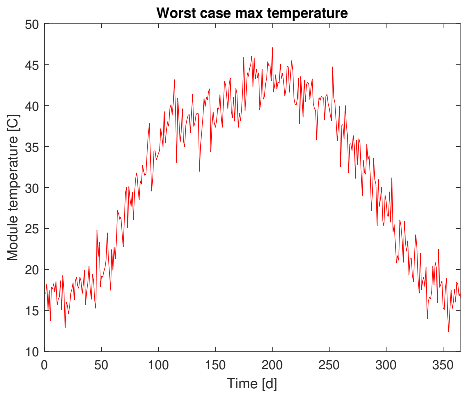


Figure 6.16: Worst case maximum temperature

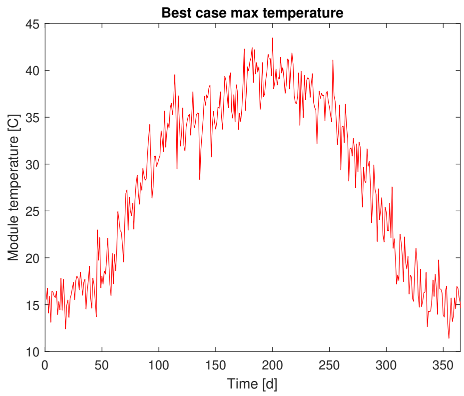


Figure 6.17: Best case maximum temperature

6.5. Housing

The current design for the housing of the electronics can be seen in figures 6.18-6.21 with the top removed to allow for a better view of the insides. The assigned dimensions and feature names can be seen in table 6.4 and 6.5 respectively. As shown in figure 6.20, the temperature/humidity sensor and the PM sensor each have their own support. As they had the most needs, it was important to make sure they would have good enough working conditions. The temperature/humidity sensor is in direct access with the outside through a hole, which also conveniently contains the noise sensor due to its small size, and to help the airflow, there is only one other main connection to the outside at the bottom of the box, which is also used as the exit for moisture. The air input and output are separated by an extrusion to minimise the air flow between them, but because there is no direct contact with the outside to optimise the air flow in the box this could hamper the performance of the PM sensor, since the PM sensor should measure as much outside air as possible. The exact effect on the performance of the PM sensor would need to be tested with a physical prototype.
The pressure sensor and the other electronics are placed in the other compartment, since they do not need any special attention. When the 50mm thick lid at the top is added, the final dimensions of the box are 200x200x200mm, thus easily meeting the maximum size requirement of the box. However, no physical prototype of this design has been made yet, thus no tests could be made to check the performance of this design.

Table 6.4: Lengths corresponding to the letters of figures 6.18-6.21

Letter	Length
A	200mm
B	150mm
C	50mm
D	160mm
E	130mm

Table 6.5: Features corresponding to the numbers of figures 6.18-6.21

Number	Name
1	Temperature/humidity sensor
2	PM sensor
3	Connection hole
4	Temperature/humidity sensor air hole
5	PM sensor air blocker extrusion

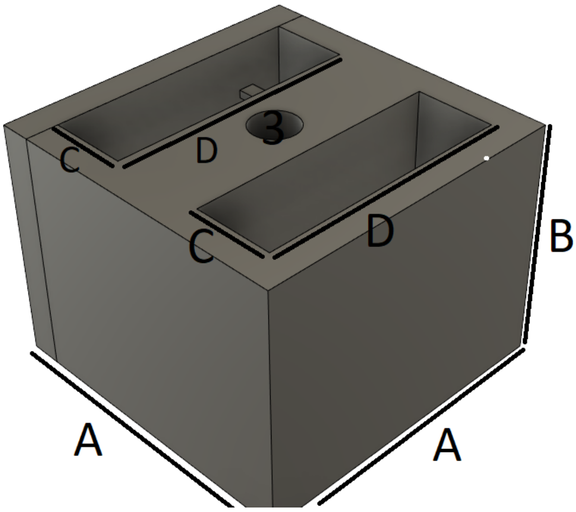


Figure 6.18: Housing neutral view with open top

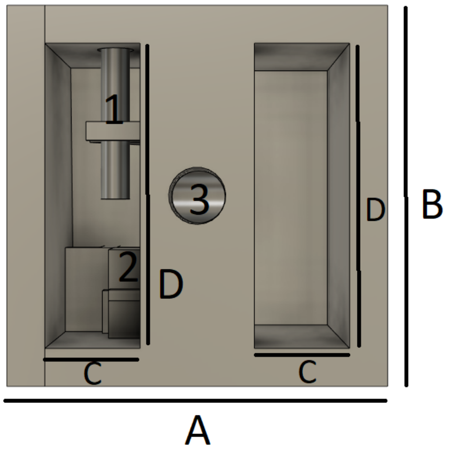


Figure 6.19: Housing top view with open top

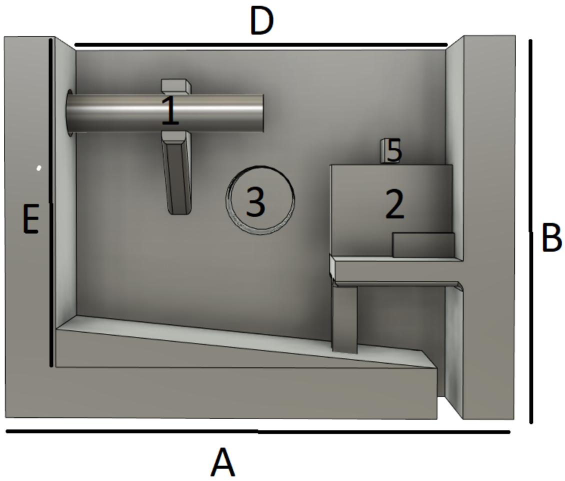


Figure 6.20: Compartment with open air contact side view

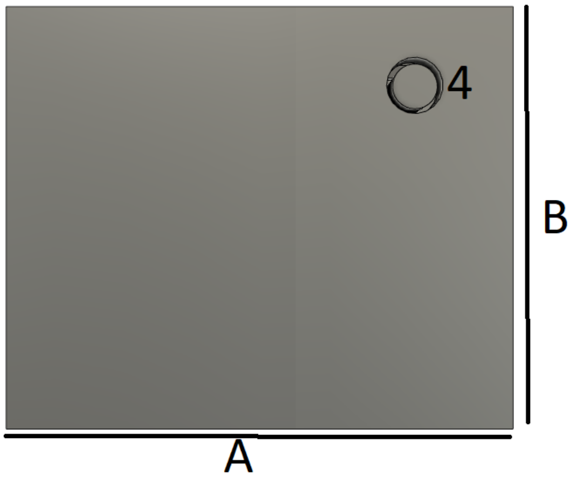


Figure 6.21: Housing side view of the temperature/humidity sensor hole

6.6. Frame

The design for the frame is not yet finalised, but the main problems have been considered. The material, aluminum, has been chosen and how it should carry/connect the box and the PV panel. Since the design it not yet finished, it could also not be made and tested. The idea on how to mount the system with a hose clamp has also been discussed. Ultimately, the trade-off requirement of having a mounting system has not been met, as the design for the frame is not finished.

7 Discussion

The results from the analysis of the digital model, based on the PV panel we have created, are promising. Figures 6.2 to 6.5 showcase that our PV panel is able to meet the requirements on the voltage set by the programme of requirements. Especially, because figures 6.16 and 6.17 show that the module temperature, where our trade-off requirement of 5V V_{mpp} is no longer met, is never actually reached. This greatly increases the performance of the PV generator. This is further corroborated by the analysis on figures 6.8, 6.10 and 6.12. These figures are the worst cases of our approximation with our digital Matlab model. They show that even in such a worst case scenario, the PV generator in conjunction with the battery is sufficient to power the system for the whole testing data set.

Furthermore, figures 6.6 and 6.7 show that the angle of the PV panel does not need to be very precise, in order for the PV generator to generate enough power. This is very important as minor differences will always occur. Additionally, these figures also prove that the optimal angle for the whole year is 38 degrees. As the additional generation in the winter months when using the winter angle, is not very vital to the performance of the weather station as shown in figure 6.8. While the over generation during the summer, from the summer angle is currently completely wasted. When comparing figures 6.8 and 6.12, the battery is just as charged during the summer. So all the extra generation shown in figure 6.6 for the summer angle compared to the winter angle during the summer, does not add any additional performance for the PV generator.

The lack of a more complex cooling management system, does not affect the performance in a noticeable way. The module temperatures at which our PV Panel would start to drop significantly in performance are never reached, according to our model. Which means that the open back of the PV Panel, and the use of convection from air flow, is sufficient as thermal management for the requirements of this project.

Key components for the effective performance of the sensors, used by the arduino, are integrated into the design of the electronic housing we have devised. The direct contact with air, for the ambient temperature sensor, is present. Likewise the PM sensor needs direct air flow through its input, while minimising the air flow from its input to its output to make sure past measurements do not affect present or future measurements. And also ensuring that different measurements take place, instead of essentially repeating the same measurement over and over. Finally, all the electronics and sensors that need to be kept away from moisture and air contact are separated from those that do need to be in contact with moisture and air. Unfortunately, there has not yet been enough time to print and/or test the effectiveness of the housing design.

Finally, although no frame is finished. There is an idea for the design. The building material of the frame, aluminium, has also been decided. As it is more readily available, as well as lighter, which is useful for the prototype.

8 Conclusion

The designed PV generator, according to the testing and analysis that have been done, meets the requirements from the programme of requirements. The housing design and the draft for the frame have been made, but not yet implemented. This means that the housing design and frame still need to be tested and optimised.

Future progress on the PV generator would involve a number of things. Further testing of the PV panel, as currently there is too much generation during the summer. So it is important to try and find a minimum requirement for the type, size and amount of solar cells, while keeping a sufficient V_{mpp}, possibly by further cutting of the cells. This might also include further research into the validity of each of the covered angles for the PV panel, as currently a case can be made for each of them. Therefore their advantages and disadvantages need to be further explored. This could involve putting the additional generation during the summer to better use with real time measuring and sending of data or the system could be made to be more minimalist using the extra generation during the winter.

Additionally, an optimisation of the design of the housing is necessary, which can only follow from thorough testing of said housing. The same is true for the frame of the PV generator. Finally, although the digital model suggests that the current PV generator design is sufficient for our needs, no physical testing of it has been done aside from one standard test conditions measurement and one outside test of the PV panel. Which means the design has not yet been proven to be sufficient in reality. This would be the final and ultimate test for this prototype.

A

Figures

A.1. Yearly generation averaged over 12 years

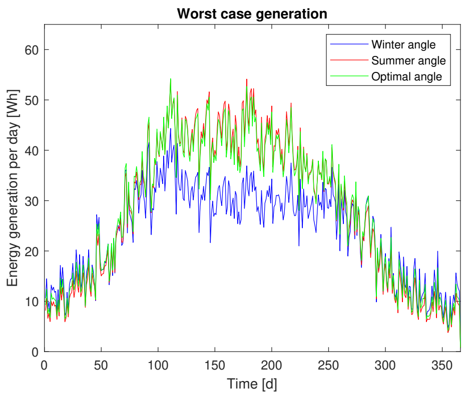


Figure A.1: The total generation per day for a year

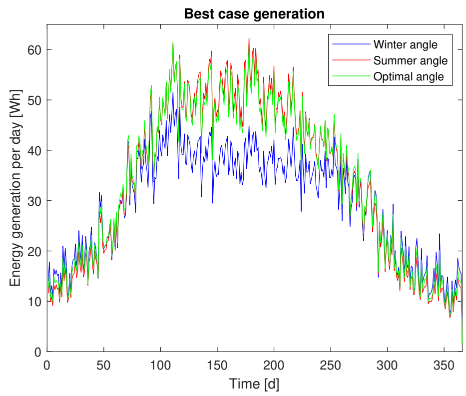


Figure A.2: The total generation per day for a year

A.2. Battery performance over 12 years

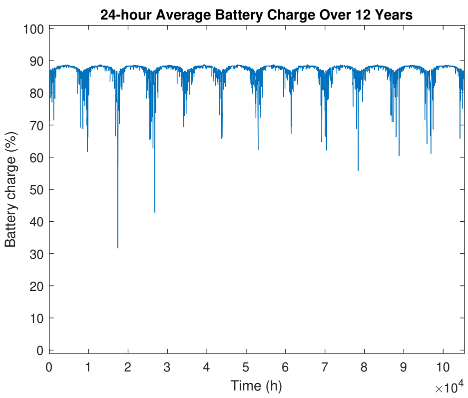


Figure A.3: Worst case battery charge with winter angle

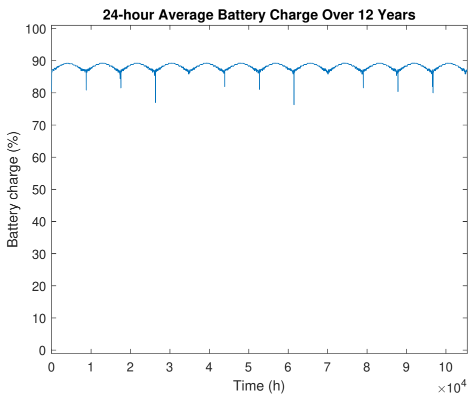


Figure A.4: Best case battery charge with winter angle

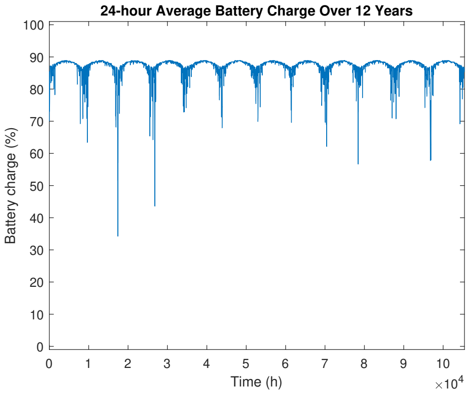


Figure A.5: Worst case battery charge with optimal angle

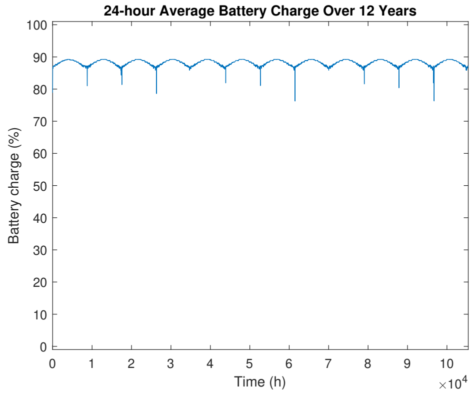


Figure A.6: Best case battery charge with optimal angle

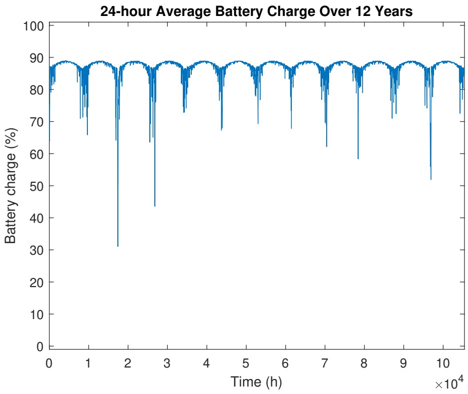


Figure A.7: Worst case battery charge with summer angle

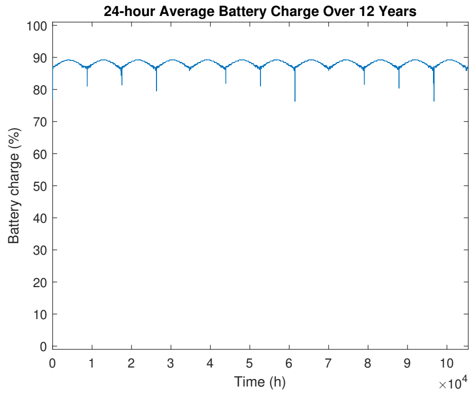


Figure A.8: Best case battery charge with summer angle

B Code

B.1. MATLAB code

[19]

B.1.1. Irradiance Analysis

```
%Student: Morten Haas, 4701852
%Student: Aru van der Knaap, 4693531
%Date: 01/06/2021
clear all;


data = readmatrix("Delft_Irradiance_2005-2016.csv");
data = data(1:105192,2:7);
%sort data to hours for each year
y = 12;
n = 8760; %normal year
s = 8784; %gap year
hc = 0; %total hour counter
for i = 1:y
    if floor(i/4)==i/4
        for j = 1:s
            yearly_data(j,i,:) = data(hc+j,:);

        end
        hc = hc + 8784;
    else
        for j = 1:n
            yearly_data(j,i,:) = data(hc+j,:);

        end
        hc = hc + 8760;
    end
end


%sort data to hours for each day
d = 366;
h = 24;
for i = 1:d
    for j = 1:h
```

```
        daily_data(j,i,:,:) = yearly_data(j+(i-1)*h,:,:);
    end
end

%Data Analysis
%Sum of daily irradiance data
%Mean of daily data
%Longest period of low sunlight
%Sum of hourly irradiance each day
daily_data_sum(:,:,:) = sum(daily_data(:,:,:,1:3),1);
%Mean of daily data
daily_data_mean(:,:,:) = mean(daily_data,1);

%plot daily global direct irradiance
% plot(daily_data_mean(:,1,1))

        3. counter increase when under threshold, restart when above
         ↪ threshold. Save all counters and locations in an array.
clear low_irradiance_mean
threshold_mean = 5;
counter = 0;
n = 1;
for i = 1:12
    for j = 1:366
        if daily_data_mean(j,i,1)<=threshold_mean
            counter = counter + 1;
        end
        if daily_data_mean(j,i,1)>threshold_mean && counter>0
            low_irradiance_mean(n,:) = [counter,j,i];
            counter = 0;
            n = n+1;
        end
    end
end
% longest_low_irradiance_mean = max(low_irradiance_mean(:,1))

%    4. production = efficiency*hourly irradiance*area factor
%        consumption = constant per hour
%        battery capacity = Wh = current + production - consumption (per
 ↪ hour)
efficiency = 0.10;
area_factor = 0.0625; %m2
production = efficiency*area_factor*(data(:,1)+data(:,2)+data(:,3)); %W

consumption = 0.1; %W
b=2; %amount of batteries
batt_cap_max = 7.2*b; %Wh
batt_cap(1) = 0.9*batt_cap_max;
Death_counter = 0;
for i = 1:length(data(:,1))
    batt_cap(i+1) = batt_cap(i) + production(i) - consumption;
    if batt_cap(i+1) > batt_cap_max
        batt_cap(i+1) = batt_cap_max;
    elseif batt_cap(i+1) <= 0
        Death_counter = Death_counter + 1;
        i
```

```
        batt_cap(i+1) = batt_cap_max;
    else
        batt_cap(i+1) = batt_cap(i+1);
    end
end
Death_counter

%plot battery capacity
plot(batt_cap)
title(['Battery Capacity Over Time For ',num2str(b), ' Batteries'])
xlim([0 length(data(:,1))])
ylim([0 batt_cap_max])
xlabel('Time(h)')
ylabel('Capacity(Wh)')

%plot generation and consumption
plot(production)
title('Production And Consumption')
xlim([0 length(production)])
xlabel('Time(h)')
ylabel('Power(W)')
yline(consumption,'k','Consumption')
```

B.1.2. Five parameter estimation

```
%Student: Morten Haas, 4701852
%Student: Aru van der Knaap, 4693531
%Date: 01/06/2021
%%
% 5 parameters for model:
%%
% * Ipv = photo current
% * Io = inverse saturation current
% * A = ideality factor
% * Rs = series resistance
% * Rp = shunt resistance

clear;
clc;

data = readmatrix('IV_trace.xlsx');
IV_data = data(:,1:3);
% first column = voltage, second column = current, third column = power

IV_parameters = data(1:5,9);
Voc = IV_parameters(1);
Isc = IV_parameters(2);
Pmax = IV_parameters(3);
Vmmp = IV_parameters(4);
Immp = IV_parameters(5);

N = 12; %number of cells
%%
% Finding the characteristic parameters
%
```

```
% Step 1: A & Rs

A = 1;
tol = 0.1;
iter = 10000;

% calculate vt, Rs, Io and Ipv

vt = vt_est(A,N);
Rs = Rs_est(Voc,Vmmp,Immp,vt);
Io = Io_est(Voc,Isc,vt,Rs);
Ipv = Ipv_est(Voc,vt,Io);

Vmpc = Vmpc_est(Immp,vt,Rs,Io,Ipv);

for i=0:iter
    if Vmpc > Vmmp
        A = A + 0.01;
    else
        A = A - 0.01;
    end

    vt = vt_est(A,N);
    Io = Io_est(Voc,Isc,vt,Rs);
    Ipv = Ipv_est(Voc,vt,Io);
    Vmpc = Vmpc_est(Immp,vt,Rs,Io,Ipv);

    err = abs(Vmpc - Vmmp);

    if err < tol
        break
    end
end

vt = vt_est(A,N);
Rs = Rs_est(Voc,Vmpc,Immp,vt);
A;
%%
% Step 2: estimation of Rp

tol = 0.001;
iter = 10000;

vt = vt_est(A,N);
Io = Io_est(Voc,Isc,vt,Rs);
Ipv = Ipv_est(Voc,vt,Io);

Rp = Rp_est(Pmax,Vmmp,Immp,Rs,Ipv,Io,vt);
Io = Io_new_est(Voc,Isc,Rs,Rp,vt);
Ipv = Ipv_new_est(Voc,Io,Rp,vt);

Impc = Pmax/Vmpc;

for j = 0:iter
    if Impc < Immp
        Rpnew = Rp + 0.1*j;
```

```
    elseif Impc >= Immp
        Rpnew = Rp - 0.1*j;
    end

    Io = Io_new_est(Voc,Isc,Rs,Rpnew,vt);
    Ipv = Ipv_new_est(Voc,Io,Rpnew,vt);

    Impc = I_est(Vmmp,Impc,Ipv,Io,Rs,Rpnew,vt);
    err = abs(Immp - Impc);

    if err < tol
        break
    end
end

%%
% Results

X = sprintf('A = %.2f, I0 = %d, Ipv = %.3f, Rs = %f, Rp = %f',
 ↪  A,Io,Ipv,Rs,Rpnew);
disp(X);
```

I estimation function

```
%Student: Morten Haas, 4701852
%Student: Aru van der Knaap, 4693531
%Date: 01/06/2021
function [I] = I_est(V,I,Ipv,Io,Rs,Rp,vt)
%  I_EST Function I
I = Ipv - Io*(exp((V+Rs*I)/vt)-1) - (V+Rs*I)/Rp;
end
```

I_0 estimation function

```
%Student: Morten Haas, 4701852
%Student: Aru van der Knaap, 4693531
%Date: 01/06/2021
function [Io] = Io_est(Voc,Isc,vt,Rs)
%  IO_EST Function Io
Io = Isc/(exp(Voc/vt)-exp((Rs*Isc)/vt));
end
```

Second I_0 estimation function

```
%Student: Morten Haas, 4701852
%Student: Aru van der Knaap, 4693531
%Date: 01/06/2021
function [Io] = Io_new_est(Voc,Isc,Rs,Rp,vt)
%  IO_NEW_EST Function new Io
Io = (Isc*(1+Rs/Rp)-Voc/Rp)/(exp(Voc/vt)-exp((Rs*Isc)/vt));
end
```

I_{pv} estimation function

```
%Student: Morten Haas, 4701852
%Student: Aru van der Knaap, 4693531
%Date: 01/06/2021
function [Ipv] = Ipv_est(Voc,vt,Io)
%  IPV_EST Function Ipv
Ipv = Io*(exp(Voc/vt)-1);
end
```

Second I_{pv} estimation function

```
%Student: Morten Haas, 4701852
%Student: Aru van der Knaap, 4693531
%Date: 01/06/2021
function [Ipv] = Ipv_new_est(Voc,Io,Rp,vt)
%  IPV_NEW_EST Function new Ipv
Ipv = Io*(exp(Voc/vt)-1)+Voc/Rp;
end
```

R_s estimation function

```
%Student: Morten Haas, 4701852
%Student: Aru van der Knaap, 4693531
%Date: 01/06/2021
function [Rs] = Rs_est(Voc,Vmmp,Immp,vt)
%  RS_EST Function Rs
Rs = (Voc/Immp) + (vt/Immp)*log(vt/(vt+Vmmp)) - (Vmmp/Immp);
end
```

R_p estimation function

```
%Student: Morten Haas, 4701852
%Student: Aru van der Knaap, 4693531
%Date: 01/06/2021
function [Rp] = Rp_est(Pmax,Vmmp,Immp,Rs,Ipv,Io,vt)
%  RP_EST Function Rp
Rp = (Vmmp*(Vmmp+Rs*Immp))/(Vmmp*Ipv-Vmmp*Io*(exp((Vmmp+Rs*Immp)/vt)-1)-
 ↪  Pmax);
end
```

B.1.3. Solar cell model

```
%Student: Morten Haas, 4701852
%Student: Aru van der Knaap, 4693531
%Date: 18/06/2021
% % key parameters
function [Pmax, Vmax, Imax, T] = Solar_Cell_Model_Function(G,Ta)
% Variables
V = 3.0:0.1:7;
Tnoct = 42; %20 is no cell heating and 48 is average

a_stc = 1.14;
I0_stc = 1.546782e-9;
Ipv_stc = 1.7623;
Rs_stc = 0.275408;
Rp_stc = 58.331508;
Voc_stc = 7.3;
Isc_stc = 1.754;
Vmpp_stc = 5.758;
Impp_stc = 1.5848;
k = 1.380653*10^(-23);
q = 1.60217646*10^(-19);
Kv = -0.356; % percent/celcius
Ki = 0.024; % percent/celcius
C_cell = 355.8240;
a1 = 7.021*10^(-4);
b = 1108;
N = 1;
T_stc = 298;
```

```
G_stc = 1000;
%%
% % Main formulas
T = Ta+(Tnoct-20)/800*G+273;
Rs = Rs_stc;
Rp = Rp_stc*G_stc./G;
Ipv = Ipv_stc*(1+Ki/100*(T-T_stc))*G/G_stc;
Isc = Isc_stc*G./G_stc+Ki/100*Isc_stc*(T-T_stc);
vt_stc = a_stc*k*T*N/q;
dV = vt_stc*log(G/G_stc);
Voc = Voc_stc+Kv/100*Voc_stc*(T-T_stc)+dV;
I0 = C_cell.*T.^3.*exp(-(q/k).*(1.1557-(a1*T.^2)./(T+b))./T);
a = q./(k.*T).*Voc./log(Isc./I0);
%%
% % Simulation formulas
for j = 1:length(V)
    syms I
    eqn = I ==
     ↪  Isc-I0*(exp((V(j)+I*Rs)/((a/a_stc)*vt_stc))-1)-(V(j)+I*Rs)/Rp;
    Curr(j) = double(solve(eqn,I));
end
%%
% % Curve parameters
P = Curr.*V;
[Pmax,Indexmax] = max(P);
Vmax = V(Indexmax);
Imax = Curr(Indexmax);
% end
end
```

B.1.4. Solar Panel LUT, LTspice

```
%Student: Morten Haas, 4701852
%Student: Aru van der Knaap, 4693531
%Date: 18/06/2021
%Improved spice model
clear all;

%import data
panel_65 = readmatrix("Single_cell_panel_T_65");
panel_45 = readmatrix("Single_cell_panel_T_45");
panel_25 = readmatrix("Single_cell_panel_T_25");
panel_5 = readmatrix("Single_cell_panel_T_5");

[parameter_65, data_65] = solar_ltspice_parameters(panel_65);
[parameter_45, data_45] = solar_ltspice_parameters(panel_45);
[parameter_25, data_25] = solar_ltspice_parameters(panel_25);
[parameter_5, data_5] = solar_ltspice_parameters(panel_5);



%Comparison of models

for i=1:10
    PmaxL_65(i) = max(data_65(:,3*i));
end
for i=1:10
```

```
    PmaxL_45(i) = max(data_45(:,3*i));
end
for i=1:10
    PmaxL_25(i) = max(data_25(:,3*i));
end
for i=1:10
    PmaxL_5(i) = max(data_5(:,3*i));
end

%Matlab data
load('Panel_Simulation_Data_G_338.mat',"PmaxG","IndexmaxG");
PmaxM_65 = PmaxG;
Pindex_65 = IndexmaxG;
load('Panel_Simulation_Data_G_318.mat',"PmaxG","IndexmaxG");
PmaxM_45 = PmaxG;
Pindex_45 = IndexmaxG;
load('Panel_Simulation_Data_G_298.mat',"PmaxG","IndexmaxG");
PmaxM_25 = PmaxG;
Pindex_25 = IndexmaxG;
load('Panel_Simulation_Data_G_278.mat',"PmaxG","IndexmaxG");
PmaxM_5 = PmaxG;
Pindex_5 = IndexmaxG;

%temperature difference
for i=1:10
    DiffL_65(i) = ((PmaxL_65(i)-PmaxL_25(i))/PmaxL_25(i))/40*100;
    DiffL_45(i) = ((PmaxL_45(i)-PmaxL_25(i))/PmaxL_25(i))/20*100;
    DiffL_5(i) = ((PmaxL_5(i)-PmaxL_25(i))/PmaxL_25(i))/-20*100;
    DiffM_65(i) = ((PmaxM_65(i)-PmaxM_25(i))/PmaxM_25(i))/40*100;
    DiffM_45(i) = ((PmaxM_45(i)-PmaxM_25(i))/PmaxM_25(i))/20*100;
    DiffM_5(i) = ((PmaxM_5(i)-PmaxM_25(i))/PmaxM_25(i))/-20*100;
end

DiffL = (sum(DiffL_65)+sum(DiffL_45)+sum(DiffL_5))/30

DiffM = (sum(DiffM_65)+sum(DiffM_45)+sum(DiffM_5))/30


% Comparison plots
V = 0.1:0.1:9;

%plot of T=65
load('Panel_Simulation_Data_G_338.mat')
for i=1:10
    p = plot(data_65(:,3*i-2),data_65(:,3*i-1));
    hold on;
end

for i=1:10
    b = plot(V,CurrG(:,i), '--');
end
hold off;
title(['I-V curves with T = 65 degrees celsius '])
xlabel(['Voltage [V]'])
ylabel(['Current [A]'])
ylim([0 2])
```

```
legend([p b],{'LTSPice','Matlab'})

%plot of T=45
load('Panel_Simulation_Data_G_318.mat')
for i=1:10
    p = plot(data_45(:,3*i-2),data_45(:,3*i-1));
    hold on;
end

for i=1:10
    b = plot(V,CurrG(:,i), '--');
end
hold off;
title(['I-V curves with T = 45 degrees celsius '])
xlabel(['Voltage [V]'])
ylabel(['Current [A]'])
ylim([0 2])

legend([p b],{'LTSPice','Matlab'})

%plot of T=25
load('Panel_Simulation_Data_G_298.mat')
for i=1:10
    p = plot(data_25(:,3*i-2),data_25(:,3*i-1));
    hold on;
end

for i=1:10
    b = plot(V,CurrG(:,i), '--');
end
hold off;
title(['I-V curves with T = 25 degrees celsius '])
xlabel(['Voltage [V]'])
ylabel(['Current [A]'])
ylim([0 2])

legend([p b],{'LTSPice','Matlab'})


%plot of T=5
load('Panel_Simulation_Data_G_278.mat')
for i=1:10
    p = plot(data_5(:,3*i-2),data_5(:,3*i-1));
    hold on;
end

for i=1:10
    b = plot(V,CurrG(:,i), '--');
end
hold off;
title(['I-V curves with T = 5 degrees celsius '])
xlabel(['Voltage [V]'])
ylabel(['Current [A]'])
ylim([0 2])
```

```
legend([p b],{'LTSPice','Matlab'})
```

B.1.5. Solar model analysis

```
%Student: Morten Haas, 4701852
%Student: Aru van der Knaap, 4693531
%Date: 18/06/2021
clear all;

%load Pmax LUT
load('Solar_LUT.mat');

%load weather data
Data14 = readmatrix('Delft_Irradiance_76deg.csv');
Data38 = readmatrix('Delft_Irradiance_2005-2016.csv');
Data61 = readmatrix('Delft_Irradiance_29deg.csv');

%data processing
[~,G14,Ta14] = Weather_Data_Processing(Data14);
[~,G38,Ta38] = Weather_Data_Processing(Data38);
[~,G61,Ta61] = Weather_Data_Processing(Data61);

%model input
years = 12;
d = 366;
hours = 24*d;
Gstep = 50;
Tstep = 5;
for j = 1:years
    for i = 1:hours
        [Pmax14(i,j),Vmax14(i,j),Imax14(i,j),T14(i,j)] = So-
         ↪ lar_Power_Round(G14(i,j),Ta14(i,j),Gstep,Tstep,0,Solar_LUT,Solar_V_LUT,Sola
        [Pmax14B(i,j),Vmax14B(i,j),Imax14B(i,j),T14B(i,j)] = So-
         ↪ lar_Power_Round(G14(i,j),Ta14(i,j),Gstep,Tstep,1,Solar_LUT,Solar_V_LUT,Sola

        [Pmax38(i,j),Vmax38(i,j),Imax38(i,j),T38(i,j)] = So-
         ↪ lar_Power_Round(G38(i,j),Ta38(i,j),Gstep,Tstep,0,Solar_LUT,Solar_V_LUT,Sola
        [Pmax38B(i,j),Vmax38B(i,j),Imax38B(i,j),T38B(i,j)] = So-
         ↪ lar_Power_Round(G38(i,j),Ta38(i,j),Gstep,Tstep,1,Solar_LUT,Solar_V_LUT,Sola

        [Pmax61(i,j),Vmax61(i,j),Imax61(i,j),T61(i,j)] = So-
         ↪ lar_Power_Round(G61(i,j),Ta61(i,j),Gstep,Tstep,0,Solar_LUT,Solar_V_LUT,Sola
        [Pmax61B(i,j),Vmax61B(i,j),Imax61B(i,j),T61B(i,j)] = So-
         ↪ lar_Power_Round(G61(i,j),Ta61(i,j),Gstep,Tstep,1,Solar_LUT,Solar_V_LUT,Sola
    end
end

% Total and Diff Power
mode = 1;
if mode == 1
% average generation per hour of the year between the 12 years
for i=1:hours
    P14(i) = sum(Pmax14(i,:))/12;
    P14B(i) = sum(Pmax14B(i,:))/12;
    P38(i) = sum(Pmax38(i,:))/12;
    P38B(i) = sum(Pmax38B(i,:))/12;
    P61(i) = sum(Pmax61(i,:))/12;
```

```
    P61B(i) = sum(Pmax61B(i,:))/12;
end

% sum generation per day, to make figures understandable
for i=1:d
    Pd14(i) = sum(P14(24*i-23:24*i));
    Pd14B(i) = sum(P14B(24*i-23:24*i));
    Pd38(i) = sum(P38(24*i-23:24*i));
    Pd38B(i) = sum(P38B(24*i-23:24*i));
    Pd61(i) = sum(P61(24*i-23:24*i));
    Pd61B(i) = sum(P61B(24*i-23:24*i));
end
Pd14 = Pd14';
Pd14B = Pd14B';
Pd38 = Pd38';
Pd38B = Pd38B';
Pd61 = Pd61';
Pd61B = Pd61B';

elseif mode == 2

for j=1:years
    for i=1:d
        Pd14(i,j) = sum(Pmax14(24*i-23:24*i,j));
        Pd14B(i,j) = sum(Pmax14B(24*i-23:24*i,j));
        Pd38(i,j) = sum(Pmax38(24*i-23:24*i,j));
        Pd38B(i,j) = sum(Pmax38B(24*i-23:24*i,j));
        Pd61(i,j) = sum(Pmax61(24*i-23:24*i,j));
        Pd61B(i,j) = sum(Pmax61B(24*i-23:24*i,j));
    end
end

end

% for mode = 1

%worst case
plot(Pd14,'blue')
hold on;
plot(Pd61,'red')
plot(Pd38,'green')
hold off;
title(['Worst case generation'])
ylabel(['Energy generation per day [Wh]'])
xlabel(['Time [d]'])
xlim([0 366])
ylim([0 65])
legend('Winter angle', 'Summer angle', 'Optimal angle')

%best case
plot(Pd14B,'blue')
hold on;
plot(Pd61B,'red')
plot(Pd38B,'green')
hold off;
title(['Best case generation'])
```

```
ylabel(['Energy generation per day [Wh]'])
xlabel(['Time [d]'])
xlim([0 366])
ylim([0 65])
legend('Winter angle', 'Summer angle', 'Optimal angle')


% for mode = 2
% worst year plot

year = 2;
%worst case
plot(Pd14(:,year),'blue')
hold on;
plot(Pd61(:,year),'red')
plot(Pd38(:,year),'green')
hold off;
title(['Worst case generation'])
ylabel(['Energy generation per day [Wh]'])
xlabel(['Time [d]'])
xlim([0 366])
ylim([0 80])
legend('Winter angle', 'Summer angle', 'Optimal angle')

%best case
plot(Pd14B(:,year),'blue')
hold on;
plot(Pd61B(:,year),'red')
plot(Pd38B(:,year),'green')
hold off;
title(['Best case generation'])
ylabel(['Energy generation per day [Wh]'])
xlabel(['Time [d]'])
xlim([0 366])
ylim([0 80])
legend('Winter angle', 'Summer angle', 'Optimal angle')



%temperature plots

%average temperature per hour of the year between the 12 years
for i=1:hours
    Tav14(i) = sum(T14(i,:))/12;
    Tav14B(i) = sum(T14B(i,:))/12;
    Tav38(i) = sum(T38(i,:))/12;
    Tav38B(i) = sum(T38B(i,:))/12;
    Tav61(i) = sum(T61(i,:))/12;
    Tav61B(i) = sum(T61B(i,:))/12;
end

%average temperature per day
for i=1:d
    Td14(i) = sum(Tav14(24*i-17:24*i-6))/12;
    Td14B(i) = sum(Tav14B(24*i-17:24*i-6))/12;
    Td38(i) = sum(Tav38(24*i-17:24*i-6))/12;
```

```
    Td38B(i) = sum(Tav38B(24*i-17:24*i-6))/12;
    Td61(i) = sum(Tav61(24*i-17:24*i-6))/12;
    Td61B(i) = sum(Tav61B(24*i-17:24*i-6))/12;
end

%max temperature per day
for j=1:years
    for i=1:d
        Tm14(i,j) = max(T14(24*i-23:24*i,j));
        Tm14B(i,j) = max(T14B(24*i-23:24*i,j));
        Tm38(i,j) = max(T38(24*i-23:24*i,j));
        Tm38B(i,j) = max(T38B(24*i-23:24*i,j));
        Tm61(i,j) = max(T61(24*i-23:24*i,j));
        Tm61B(i,j) = max(T61B(24*i-23:24*i,j));
    end
end

for i=1:d
    Tma14(i) = sum(Tm14(i,:))/12;
    Tma14B(i) = sum(Tm14B(i,:))/12;
    Tma38(i) = sum(Tm38(i,:))/12;
    Tma38B(i) = sum(Tm38B(i,:))/12;
    Tma61(i) = sum(Tm61(i,:))/12;
    Tma61B(i) = sum(Tm61B(i,:))/12;
end

Tma14 = Tma14';
Tma14B = Tma14B';
Tma38 = Tma38';
Tma38B = Tma38B';
Tma61 = Tma61';
Tma61B = Tma61B';

%worst case avg
plot(Td38,'red')
hold off;
title(['Worst case average temperature'])
ylabel(['Module temperature [C]'])
xlabel(['Time [d]'])
xlim([0 366])

%best case avg
plot(Td38B,'red')
hold off;
title(['Best case average temperature'])
ylabel(['Module temperature [C]'])
xlabel(['Time [d]'])
xlim([0 366])

%worst case max
plot(Tma38,'red')
title(['Worst case max temperature'])
ylabel(['Module temperature [C]'])
xlabel(['Time [d]'])
xlim([0 366])
```

```
%best case max
plot(Tma38B,'red')
title(['Best case max temperature'])
ylabel(['Module temperature [C]'])
xlabel(['Time [d]'])
xlim([0 366])
```

B.1.6. Rounding solar power function

```
%Student: Morten Haas, 4701852
%Student: Aru van der Knaap, 4693531
%Date: 18/06/2021
function [Pmax,Vmax,Imax,T] = So-
  ↪ lar_Power_Round(G,Ta,Gstep,Tstep,R,Solar_LUT,Solar_V_LUT,Solar_I_LUT,Solar_T_LUT)
if R == 0 % worst case rounding
   Gf = floor(G/Gstep);
   Tac = ceil(Ta/Tstep);
   if Gf <= 0
        Pmax=0;
        Vmax=0;
        Imax=0;
        T=Ta;
   else
        if Tac >= 10
            Tac = 9;
        elseif Tac <= 0
            Tac = 1;
        end
        Pmax = Solar_LUT(Gf,Tac);
        Vmax = Solar_V_LUT(Gf,Tac);
        Imax = Solar_I_LUT(Gf,Tac);
        T = Solar_T_LUT(Gf,Tac);
   end
elseif R == 1 % best case rounding
   Gf = ceil(G/Gstep);
   Tac = floor(Ta/Tstep);
   if Gf <= 0
        Pmax=0;
        Vmax=0;
        Imax=0;
        T=Ta;
   else
        if Tac >= 10
            Tac = 9;
        elseif Tac <= 0
            Tac = 1;
        elseif Gf >= 25
            Gf = 25;
        end
        Pmax = Solar_LUT(Gf,Tac);
        Vmax = Solar_V_LUT(Gf,Tac);
        Imax = Solar_I_LUT(Gf,Tac);
        T = Solar_T_LUT(Gf,Tac);
   end
else
    Pmax=0;
    Vmax=0;
```

```
    Imax=0;
    T=Ta;
end


end
```

B.1.7. Creation of LUT

```
%Student: Morten Haas, 4701852
%Student: Aru van der Knaap, 4693531
%Date: 18/06/2021
clear all;

Gstep = 50;
Tstep = 5;
Gtot = 1200/Gstep;
Ttot = 45/Tstep;
tic
for j = 1:Ttot
    for i = 1:Gtot
        N = (j-1)*24+i
        [Solar_LUT(i,j),Solar_V_LUT(i,j),Solar_I_LUT(i,j),Solar_T] =
         ↪ Solar_Cell_Model_Function(Gstep*i,Tstep*j);
        Solar_T_LUT(i,j) = Solar_T - 273;
        toc
    end
end
```

B.1.8. Weather data processing

```
%Student: Morten Haas, 4701852
%Student: Aru van der Knaap, 4693531
%Date: 18/06/2021

function [yearly_data,irradiance_sum,yearly_temp] =
 ↪ Weather_Data_Processing(data)

data = data(1:105192,2:6);
data(:,4) = [];
%sort data to hours for each year
y = 12;
n = 8760; %normal year
s = 8784; %gap year
hc = 0; %total hour counter
for i = 1:y
    if floor(i/4)==i/4
        for j = 1:s
            yearly_data(j,i,:) = data(hc+j,:);

        end
        hc = hc + 8784;
    else
        for j = 1:n
            yearly_data(j,i,:) = data(hc+j,:);

        end
```

```
        hc = hc + 8760;
    end
end
yearly_temp(:,:,:) = yearly_data(:,:,4);
irradiance_sum(:,:,:) = sum(yearly_data(:,:,1:3),3);

%sort data to hours for each day
% d = 366;
% h = 24;
% for i = 1:d
%     for j = 1:h
%         daily_data(j,i,:,:) = yearly_data(j+(i-1)*h,:,:);
%     end
% end

% Data Analysis
% Sum of daily irradiance data
% daily_data_sum(:,:,:) = sum(daily_data(:,:,:,1:3),1);

end
```

B.2. LTspice code

[26]

B.2.1. single diode model

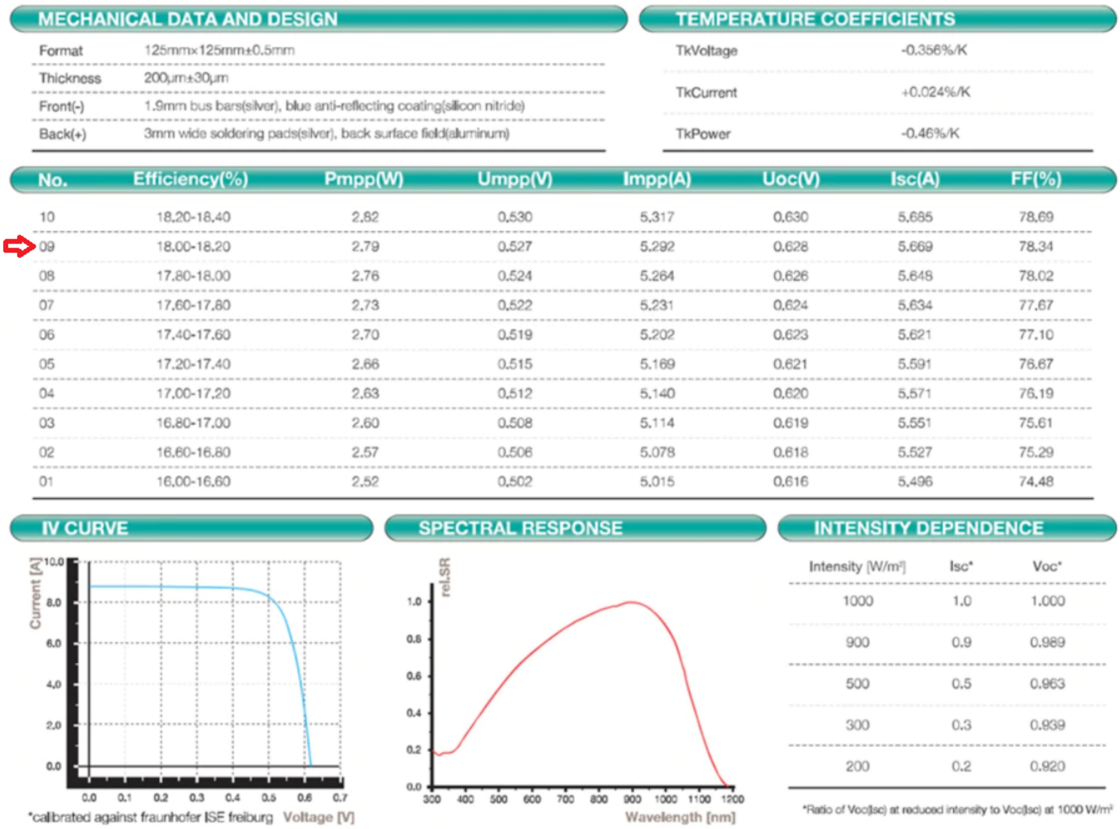
```
*Author:    Tibbe van der Biezen (4686608)
*Author:    Aru van der Knaap (4693531)
*Date:  04/06/2021
.param Isc_stc = 1.754 ;Short circuit current
+Rs_stc = 0.275408 ;Series resistance
+Rp_stc = 58.331508 ;Shunt resistance
+Voc_stc = 7.3 ;Open circuit current
+Irs = 1.546782n ;Reverse saturation current
+Ipv = 1.762  ;Photo current
+G_stc = 1000 ;Standard irradiance
+G = {Gs} ;Irradiance
+a_stc = 1.14 ;Ideality factor
+Kv = -0.356 ;percent per Celsius
+Ki = 0.024 ;percent per Celsius
+T_stc = 25 ;Temperature
+T = 5
+Vmpp_stc = 5.758 ;Voltage at mpp
+Impp_stc = 1.5848 ;Current at mpp
+k =1.380653e-23 ;Boltzman constant
+q=1.60217646e-19 ;Electric charge
+N=1  ;Number of cells

*Formulas********
+Rs = Rs_stc
+Rp = {Rp_stc*G_stc/G}
+Isc = {Isc_stc*G/G_stc+Ki/100*Isc_stc*(T-T_stc)}
+vt = {a_stc*k *T*N/q}
+dV = {vt*ln({G}/G_stc)}
+Voc={Voc_stc+Kv/100*Voc_stc*(T-T_stc)+{dV}}
+Ec = {q/(k*(T+273.15))*{Voc}/ln({Isc}/{Irs})}
```

```
.model DiodeSC D(Is={Irs} N={Ec})
```

C

Datasheets

C.1. Mystery solar cell datasheet



MECHANICAL DATA AND DESIGN

Format	125mm×125mm±0.5mm
Thickness	200μm±30μm
Front(-)	1.9mm bus bars(silver), blue anti-reflecting coating(silicon nitride)
Back(+)	3mm wide soldering pads(silver), back surface field(aluminum)

TEMPERATURE COEFFICIENTS

TkVoltage	-0.356%/K
TkCurrent	+0.024%/K
TkPower	-0.46%/K

No.	Efficiency(%)	Pmpp(W)	Umpp(V)	Impp(A)	Uoc(V)	Isc(A)	FF(%)
10	18.20-18.40	2.82	0.530	5.317	0.630	5.685	78.69
09	18.00-18.20	2.79	0.527	5.292	0.628	5.669	78.34
08	17.80-18.00	2.76	0.524	5.264	0.626	5.648	78.02
07	17.60-17.80	2.73	0.522	5.231	0.624	5.634	77.67
06	17.40-17.60	2.70	0.519	5.202	0.623	5.621	77.10
05	17.20-17.40	2.66	0.515	5.169	0.621	5.591	76.67
04	17.00-17.20	2.63	0.512	5.140	0.620	5.571	76.19
03	16.80-17.00	2.60	0.508	5.114	0.619	5.551	75.61
02	16.60-16.80	2.57	0.506	5.078	0.618	5.527	75.29
01	16.00-16.60	2.52	0.502	5.015	0.616	5.496	74.48

Intensity [W/m²]	Isc*	Voc*
1000	1.0	1.000
900	0.9	0.989
500	0.5	0.963
300	0.3	0.939
200	0.2	0.920

*Ratio of Voc(Isc) at reduced intensity to Voc(Isc) at 1000 W/m²

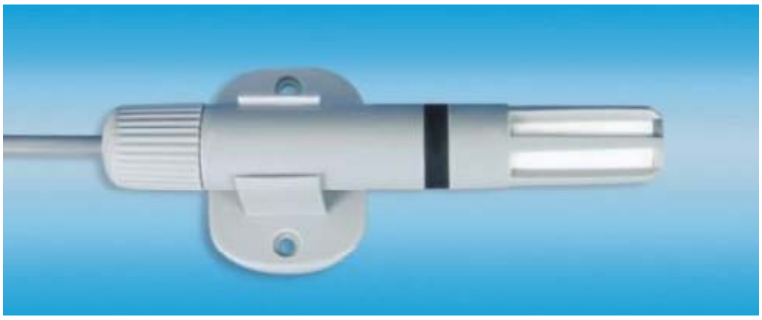
Figure C.1: Mystery solar cell datasheet

C.2. Temperature/humidity sensor

AOSONG

Digital temperature and humidity sensor

AM2315 Product Manual

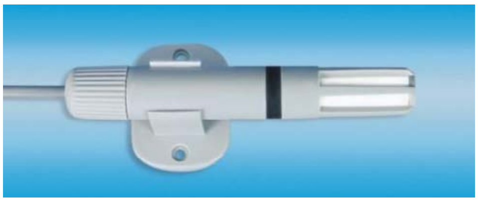


www.aosong.com

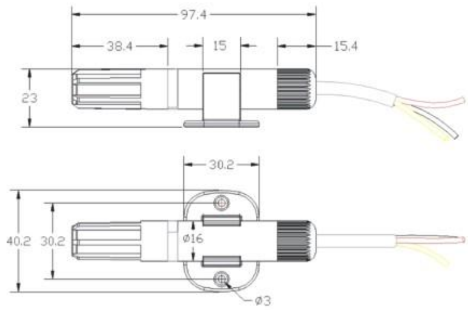
1、Product Overview

AM2315 capacitive humidity sensing digital temperature and humidity sensor is one that contains the temperature and humidity combined sensor has been calibrated digital signal output. Special temperature and humidity acquisition technology, to ensure that the product has high reliability and excellent long−term stability. The sensor includes a capacitive sensor wet components and an integrated high−precision temperature measurement devices, and connected with a high−performance microprocessor. The product has excellent quality, fast response, strong anti−jamming capability, and high cost.

AM2315 communication standard I2C communication. I2C communication standard communication timing, the user can be directly linked to on the I2C communication bus, no additional wiring, simple to use. Transmission of digital data output directly compensated by the temperature, humidity, temperature and Check (CRC) and other digital information, the user does not need the digital output for the second calculation, and no need for temperature compensation of the humidity, you can get an accurate temperature and humidity information . Product of four leads, convenient connection, special packages according to user needs.



Physical map



Dimensions (unit: mm)

2、Applications

HVAC air conditioners, dehumidifiers, testing and inspection equipment, consumer goods, automotive, automation, data loggers, weather stations, home appliances, humidifiers, medical, and other related humidity measurement and control.

3、Features

Completely interchangeable, low cost, long−term stability, relative humidity and temperature measurement, long distance signal transmission, digital signal output, precise calibration, low power consumption, standard I^2C bus digital interface.

4、Interface definition

4.1 AM2315 pin assignment

Table 1: AM2315 pin assignment

Pin	Color	Name	Description
1	Red	VDD	Power (3.5V−5.5V)
2	Yellow	SDA	Serial data, bidirectional
3	Black	GND	Ground
4	White	SCL	Serial Clock, input

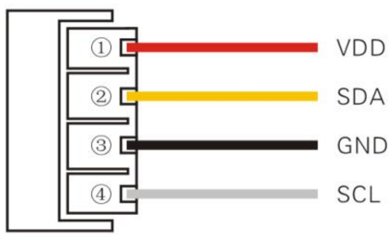


Figure 1: AM2315wiring diagram

4.2 Power supply pins（VDD GND）

AM2315 supply voltage range of 3.5 - 5.5V, recommended supply voltage is 5V.

4.3 Data line（SDA SCL）

SCL I^2C communication signal line when the clock line, SCL used for communication between the microprocessor and the AM2315 synchronous. SDA pin is tri−state structure for reading, writing sensor data. Specific communication timing, see a detailed description of communication protocols.

5、Sensor Performance

5.1 Relative Humidity

Table 2: AM2315 relative humidity performance table

Parameter	Condition	min	typ	max	Unit
Resolution			0.1		%RH
			16		bit
Accuracy [1]	25℃		±2		%RH
Repeatability			±0.1		%RH
Exchange		Completely interchangeable			
Response [2]	1/e(63%)		<5		S
Sluggish			<0.3		%RH
Drift [3]	Typical		<0.5		%RH/yr

5.2 Temperature

Table 3: AM2315 relative temperature performance table

Parameter	Condition	min	typ	max	Unit
Resolution			0.1		℃
			16		bit
Accuracy			±0.1	±1	℃
Range		−40		125	℃
Repeat			±0.2		℃
Exchange		Completely interchangeable			
Response	1/e(63%)		<5		S
Drift			±0.1		℃/yr

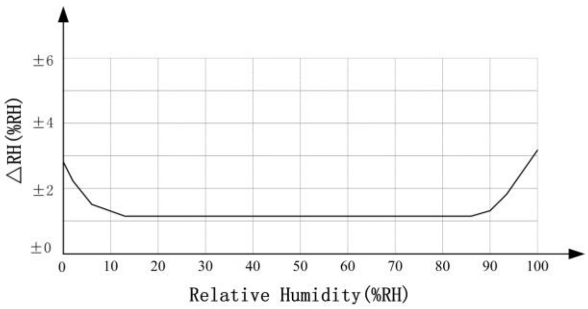


Figure 2: 25 ℃ relative humidity when the AM2315 maximum error

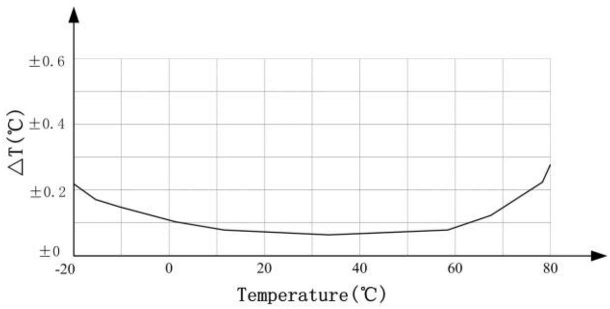


Figure 3: The maximum temperature error of temperature sensor

6、Electrical Characteristics

Electrical characteristics, such as energy consumption, high, low, input, output voltage, depends on power. Table 4 details the AM2315 electrical characteristics, if not stated otherwise supply voltage of 5V. For best results with the sensors, design strictly in accordance with Table 4 and Table 6 of the conditions of the design.

Table 4: AM2315 DC Characteristics。

Parameter	Condition	min	typ	max	Unit
Supply voltage		3.5	5	5.5	V
Power [4]	Dormancy	10	15		μA
	Measuring		500		μA
	Average		300		μA
Low-level output	I_{OL} [5]	0		300	mV
High output	Rp<25 kΩ	90%		100%	VDD
Low input voltage	Decline	0		30%	VDD
Input High	Rise	70%		100%	VDD
Rpu[6]	VDD = 5V VIN = VSS	30	45	60	kΩ
Output current	On		8		mA
	Off	10	20		μA
Sampling period		2			S

[1] The accuracy of the factory test, the sensor at 25 ℃ and 5V, the accuracy of indicators under the conditions tested, it does not include hysteresis and non-linear, and only for non-condensing environment.
[2] at 25 ℃ and 1m / s air flow conditions, to reach 63% of first-order response time required.
[3] in volatile organic compounds, the values may be higher. See manual application to store information.
[4] This value is VDD = 5.0V, when the temperature is 25 ℃, 2S / time, average conditions.
[5] low output current.
[6] that pull-up resistor.

7、I^2C Communication protocol

7.1 I^2C communication protocol introduced

7.1.1 I^2C About the bus

AM2315 and micro-processor control interface form of the I^2C serial bus, in this brief introduction about the I^2C bus protocol standard. Due to space limitations, the agreement can not list the entire contents of a deeper problem, please refer to the relevant information (refer to the Philips site inspection).

7.1.2 I^2C Bus Overview

Philips (Philips) at 20 years ago invented a simple two-way two-wire serial communication bus, the bus is known as the Inter-I^2C bus. I^2C bus has now become the industry standard solution for embedded applications, is widely used in variety of microcontroller-based professional, consumer and telecommunications products, as a control, diagnostics and power management bus. More in line with standard I^2C bus devices can communicate with an I^2C bus, without the need to address decoder.

I^2C bus only composed by the two signal lines, a serial data line SDA, the other root is the serial clock line SCL. I^2C bus devices generally have their SDA and SCL pins are open-drain (or open collector) output structure. Therefore, the actual use, SDA and SCL signal lines must be pull-up resistor (Rp, Pull-Up Resistor). Pull-up resistor on the general value of 3 ~ 10 kΩ . Therefore, when the bus is free, the two signal lines remain high, almost no current consumption; electrical compatibility, and supports a variety of different voltage logic interface; different between the two can be directly connected to the bus, not require additional conversion circuitry to support a variety of ways a master multi-slave communication is the most common means of communication. It also supports dual-host communications, multi-host communications, and broadcast mode, and so on.

I^2C typical configuration is shown in Figure 4。

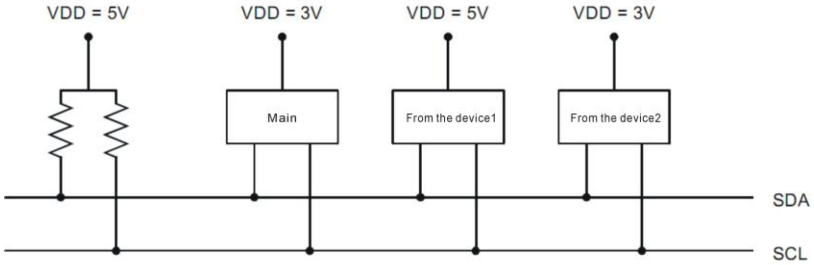


Figure 4： I^2C typical configuration

7.1.3 I^2C Bus protocol specification

◎ **I^2C Bus Definition of Terms**

I^2C−bus serial data (SDA) and serial clock (SCL) line connected to the bus, allowing the device to transmit information between each device has a unique address identification, and can be used as a transmitter or receiver (the device features decision), the device performs data transmission can also be seen as a master or a slave, the host is initialized bus allows data transfer and generates the clock signal transmission device. At this point, any device addressed is considered a slave. Details of the I^2C bus definition of terms in Table 5.

◎ **I^2C Bus transfer rate**

I^2C Bus communication speed controlled by the host, to fast to slow. However, the maximum rate is limited, I^2C bus data transfer rate up to the standard mode, fast 100Kb /s.

◎ **I^2C Bus bit transmission**

I^2C Bus bit transmission is through the data line SDA and SCL line to complete the two lines together. In the high period of SCL clock line, data line SDA low logic level, said current transmission "0"; in the high period of SCL clock line, data line SDA high logic level, said current transmission "1." Logic "0" (low) and "1" (high) level, is related by the VDD voltage level (for details see Table 4 AM2315 DC Characteristics table). In addition, each data bit transferred on to generate a clock pulse.

Table 5： I^2C Bus definitions of terms

Term	Description
Transmitter	Devices to send data to the bus
Receiver	Device receiving data from the bus
Host	Initialization and termination of the clock signal sent sent produce devices
From machine	Addressed by the host device
Multi−host	At the same time try to control more than one host, but does not destroy the message bus.
Arbitration	It's a multiple hosts simultaneously try to control the bus but only one is allowed to control the bus and make the message is not destroyed in the process.
Synchronous	Two or more devices synchronized clock signal in the process

◎ **The validity of data**

Data line SDA of the data must be in the HIGH period of the clock remain stable. SDA data line high or low state only in the clock line SCL low time was allowed to change. But in the beginning and end of the I^2C bus when the exception (for details see Start and Stop conditions). Some other data may require the serial bus clock signal edge (rising or falling edge) is valid, but the level I^2C bus is valid. Specific timing diagram shown in Figure 5.

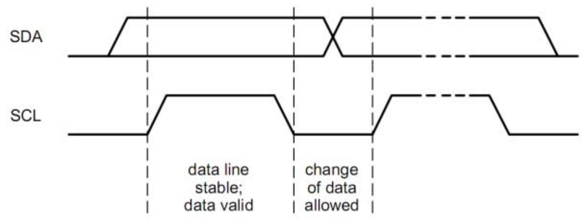


Figure 5： I^2CBus bit transmission

◎ **Start and stop conditions**

Initial conditions： During the time when SCL is high, SDA from high to low transition generated when the initial conditions. Produced in the initial conditions after the bus is considered busy. Initial conditions that are often denoted by S.

Stop condition： During the time when SCL is high, SDA from low to high transition when a Stop condition. Generate a stop condition after the bus is idle. Stop condition denoted by P.

Start and stop conditions diagram shown in Figure 6。

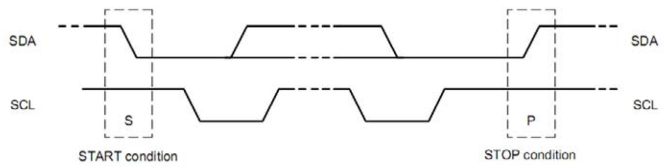


Figure 6： Schematic diagram of the start and stop conditions

◎ **Byte transfer format**

I^2C Bus to send and receive data bytes. Transferred to the SDA line must be 8-bit per byte. The number of bytes per transfer is unrestricted. First, the data transmission is the highest bit (MSB No. 7), the last transmission is the lowest bit (LSB, bit 0). In addition, each byte must be followed by an acknowledge bit (ACK). I^2C transmit data in Figure 7.

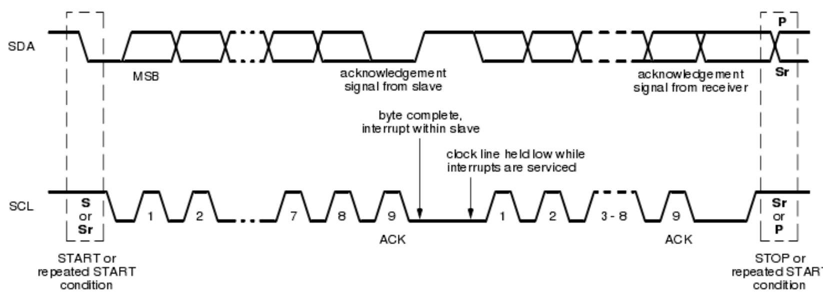


Figure 7： I^2C Bus data transfer

◎ I²C Bus response

I²C−bus transfers data in the process, each transmission of a byte, must be answered with a status bit. The case of the receiver can receive data acknowledge bit to inform the transmitter. Acknowledge bit, still the master generates the clock pulse, and an acknowledge bit, the data state is follow the "who who receives a" principle, which always generates the acknowledge bit by the receiver, in response to the clock pulse period the receiver must pull the SDA line low, making it the clock pulse is stable LOW during the HIGH period (see Figure 8), of course, setup and hold times must be considered (for details see table 6). Send data from master to slave, the slave generates the acknowledge bit; host to receive data from the slave, the acknowledge bit by the master.

I²C bus standard: 0 acknowledge bit to the receiver acknowledge (ACK), often abbreviated as A; of 1 indicates non−response (NACK), often abbreviated to NA. After the transmitter sends the LSB should release the SDA line (pulled SDA), to wait for the receiver generating an acknowledge bit.

If the receiver is receiving last byte of data, or can not receive more data, it should produce non−ACK to notify the sender. If you find the receiver transmitter generates a non−response state, you should terminate the transmission.

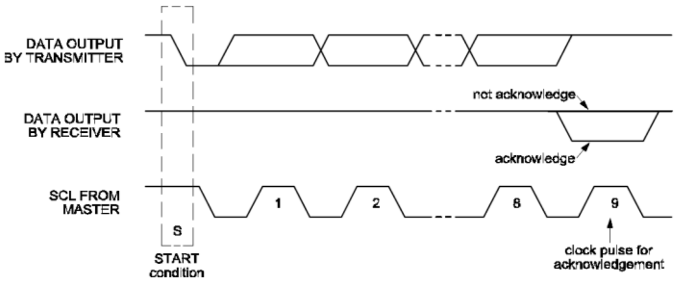


Figure 8： I²CBus response

◎ Slave Address

No additional I²C bus address decoder and chip select signals. Multiple devices with I²C bus interface can be connected to the same I²C bus, and address them through the device to distinguish between. The process of addressing the I2C bus is usually the initial conditions in the first byte after the decision to choose which one from the host machine, that is 7−bit addressing address (the other is 10−bit addressing address is different, the sensors 7−bit addressing address). The first byte of the bit definitions shown in Figure 9, the first byte of the first seven bits of the slave address, the lowest bit (LSB) is No. 8. It determines the direction of the message, the first byte of the least significant bit (LSB) is "0": Indicates that the host will write the information to be selected from the machine; "1" indicates that the master will read information from the machine.

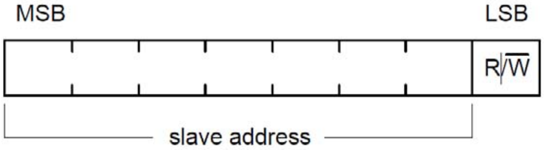


Figure 9：After the initial conditions of the first byte

Sent an address, the system of each device in the initial conditions, the first seven and more if its own address as the device will think it is the host address, as a slave receiver or slave transmitter by the R / W bits. Host is the master Parts, it does not need the device device address, and other devices are all from the machine, have the device address. Must ensure that all on the same I²C bus slave addresses are uniquely determined, can be repeated, or the I²C bus will not work.

◎ Schematic diagram of the basic data transfer format

Figure 10 and Figure 11 are given the I²C data transmit and receive the basic format. It should be noted, Figure 11 and Figure 12 is different, in Figure 11, the host sends to the last byte from the data, the response from the machine may or may not answer, but in any case the host can generate stop condition. If the host sends data to the slave (even including slave address) when the detected non-response from the machine, you should promptly stop the transmission.

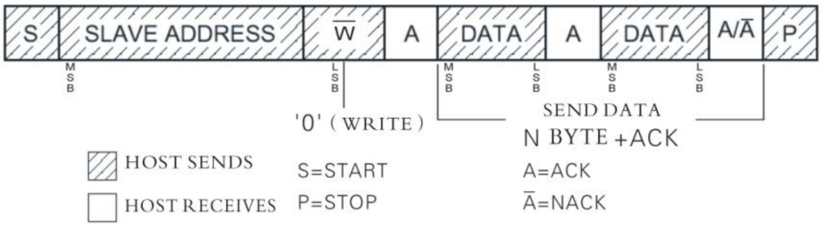


Figure 10: I²C The bus master sends data to the basic format from

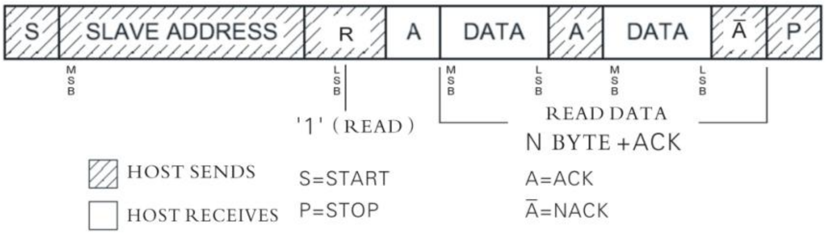


Figure 11: I²CThe bus master receives data from the slave's basic format

7.2 AM2315 Sensor I²C communication protocol

AM2315 I²C bus serial interface, in full accordance with the standard I2C protocol addressing, can be directly linked to the I²C bus. AM2315 sensor I2C address (SLAVE ADDRESS) is 0xB8, the I²C bus protocol based on the standard, based on ModBus protocol, developed a unique communication protocol, reducing the transmission error rate. Microcontroller read AM2315 sensor, the sensor I²C_ModBus in strict accordance with AM2315 communication protocol and timing design.

7.2.1 I²C Interface Description

AM2315 digital temperature and humidity sensor as a slave, and the host (user microprocessor) way communication between the I²C bus standard mode. For the accurate measurement of humidity, temperature and humidity to reduce the impact, AM2315 sensors in the non-working period, automatically be converted to sleep, to reduce the work consumption, to reduce the sensor self-heating of the humidity of the surrounding environment. AM2315 uses passive mode, that is, the host through the instruction wake-up sensor, and then sends the appropriate commands, read the corresponding values of temperature and humidity; communication after the acquisition of temperature and humidity sensor is triggered once; so if the long did not read the sensor, please read the two consecutive second sensor (minimum interval of two to read 2S), the second is the latest measured value; collected after the end of the sensor automatically be converted to sleep. Next host to be read sensor, the need to re-awaken the sensor. Must be noted that host communication from start to finish, for a maximum of 3S. If communication is not completed within 3S, sensors automatically end communication, automatically be converted to sleep, read again the host, such as sensors, need to re-send the wake-up command.

7.2.2 I²C Interface Features

This section describes the AM2315 Sensor I²C interface characteristics, if you want to get the best communications with the sensor results, designed strictly in accordance with Figure 12 and Table 6 of condition design.。

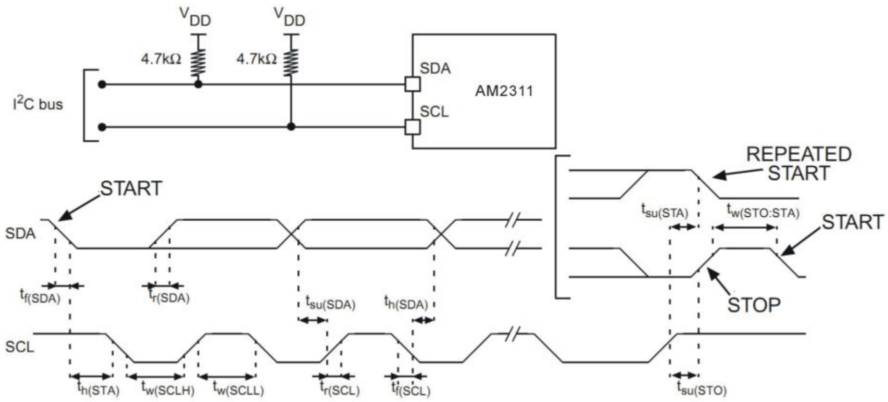


Figure 12：AM2315 Typical application circuit and the I2C bus timing diagram

Table 6：AM2315 Sensor I²C interface characteristics

Symbol	Parameters	Standard I²C mode min	Standard I²C mode max	Unit
SCLClock frequency			100	kHz
tw(SCLL)	SCLClock low time	4.7		μs
tw(SCLH)	SCLClock high time	4.0		μs
tsu(SDA)	SDA Setup time	250		ns
th(SDA)	SDA Data hold time	0[1]		ns
tr(SDA) tr(SCL)	SDA & SCL Rise time		1000	ns
tf(SDA) tf(SCL)	SDA & SCL Fall Time		300	ns
th(STA)	Start condition hold time	4.0		μs
tsu(STA)	Repeated START condition setup time	4.7		μs
tsu(STO)	Stop condition setup time	4.0		μs
tw(STO:STA)	Stop to Start condition time (bus free)	4.7		μs
Cb	Capacitive load for each bus		400	pF

7.2.3 Communication protocol

AM2315 Sensor I²C communication protocol is a standard I²C bus protocol based on the reference ModBus protocol, the sensor according to AM2315 own characteristics, a combination of I²C_ModBus agreement. The following format:：

◎ **Communication data (information frame) format**

	I^2CAdd+R/W	Function code	Data area	CRC Check [3]
Data format:				
Data length:	1Byte [2]	1Byte	NByte	16-bit CRC (cycle redundancy code)

[1] If the interface does not allow to extend the time low, only need to comply with the longest hold time Start condition.

[2] A byte consists of 8-bit binary number (both 8 bit).

[3] CRC checksum algorithm, for details see: CRC code calculation method; detailed below.

◎ **Communication and information transfer process**

When the communication command from the sending device (host) to the sensor, I^2C address of the command line with the sensor, the sensor was to receive, and in accordance with the relevant requirements of the function code and read the information; and the implementation of the results (data) back to the host. The information returned includes the function code, data and after the implementation of the CRC code (user time to read the CRC, stop conditions can be sent directly).

◎ **Communication I^2C slave address**

AM2315 I^2C addresses are the same for each sensor, and is 0xB8. Therefore, in the same bus can only be linked to a AM2315 sensor, the sensor only after receiving the start signal and the same with its I^2C address only to respond to the host.

◎ **I^2C communication function code**

Communication of information for each function code is the first byte of the frame transmission. I^2C_ModBus communication rules, define the function code is 1 to 127. As a host request, through the function code tells the slave what action should be implemented. Response as a slave, the slave returns the function code and the functions from the host to send the same code, it indicates that the response from the host machine and have been associated operations. I^2C_ModBus part of the function codes described in Table 7.

Table 7: I^2C_ModBus Part of the function code

Function code	Definition	Operations (binary)
0x03	Read register data	Read data from one or more registers
0x10	Write multiple registers	Multiple sets of binary data is written to multiple registers

◎ **I^2C communication data area**

Data area including the need to return to what information from the sensor or perform any action. This information can be data (such as: temperature, humidity, the sensor device information, user-written data, etc.), the reference address. For example, the host told the sensor 03 through the function code value return register (read register contains the starting address and length of register read), the returned data includes the length of the data register and data register contents.

I²C_Modbus sensor uses a custom communication protocol, host, use of communication commands (function code 03), can read the data register any of its data register table shown in Table 8. Sensor temperature and humidity data register stores the value and the corresponding sensor signal equipment and other related information; each data register is a single byte (8 bits) of binary data; read sensor, up to 10 registers of data, more than reading take the length of the sensor will return the corresponding error code. Error code information, see Table 1.

Table 8：AM2315 Data register table

Register information	Add	Register information	Add	Register information	Add	Register information	Add
High RH	0x00	Model high	0x08	Users register a high	0x10	Retention	0x18
Low RH	0x01	Model low	0x09	Users register a low	0x11	Retention	0x19
High temp.	0x02	Version number	0x0A	Users register 2 high	0x12	Retention	0x1A
Low temp.	0x03	ID(24−31) Bit	0x0B	Users register 2 low	0x13	Retention	0x1B
Retention	0x04	ID(16−23) Bit	0x0C	Retention	0x14	Retention	0x1C
Retention	0x05	ID(8 − 15) Bit	0x0D	Retention	0x15	Retention	0x1D
Retention	0x06	ID(0 − 7) Bit	0x0E	Retention	0x16	Retention	0x1E
Retention	0x07	Status Register	0x0F	Retention	0x17	Retention	0x1F

◎ **Temperature output format**

Temperature resolution is 16Bit, the maximum temperature position (Bit15) equal to 1 indicates a negative temperature, the temperature highest (Bit15) is equal to 0 for positive temperature; temperature in addition to highest (Bit14 ~ Bit0) that string out of the temperature sensor. String out of the temperature sensor is 10 times the actual temperature；

◎ **Status Register**

Status register, Bit7−Bit0 bit, temporarily reserved

Status register bits	Bit7	Bit6	Bit5	Bit4	Bit3	Bit2	Bit1	Bit0
Features	Retention							

◎ **I²C_ModBus Function Code Description**

一、**Function code "03"**：Read multiple sensors register

Host sends a read frame format：

START+ (I²CAddress+W)+ Function code（0x03）+ Start address + Register number +STOP

The host to read back the data：

START+ (I²C Add+R)+ Continuously read sensor data returned +STOP

Sensor response frame format：

Function code（0x03）+ Register number + Data +CRC[1]

for example：The host continues to read sensor data: 0x00 start address register of the four sensor data.

Sensor data register address and data：

Register add	Register data	Data on	Register add	Register data	Data on
0x00	0x01	High RH	0x02	0x00	High temp.
0x01	0xF4	Low RH	0x03	0xFA	Low temp.

Host sends the message format：

Host sends	Bytes	message	Remarks
Sensor add	1	0xB8	Sensor I^2C address（0xB8）+W（0）
Function code	1	0x03	Read the register
Start add	1	0x00	Register start add 0x00
Register No.	1	0x04	Read the register number

Sensor response to the return message format：

Response from machine	Bytes	Send	Remarks
Function code	1	0x03	Read the register
Returns No. of bytes	1	0x04	Back 4 of 4 bytes register
Register 1	1	0x01	Add 0x00 of the content (RH high byte)
Register 2	1	0xF4	Add 0x01 of the content (RH low byte)
Register 3	1	0x00	Add the content of 0x02 (temp. high byte)
寄存器 4	1	0xFA	Add 0x03 of the content (temp. low byte)
CRC 码	2	31A5	Back to the CRC, the low byte first ;

[1]： Details of the CRC calculations see the back of the CRC introduced, the sensor returns all the data are CRC check, the user can choose to read or not read。

Numerical：

Read back from the sensor temperature and humidity values, as long as the value into a decimal number 10 is divided by the value corresponding to the temperature and humidity, the temperature of the corresponding unit ℃, humidity units of % RH. For example, read back the data above:

Humidity: 01F4 = 1×256+15×16+4 = 500 => Humidity = 500÷10=50.0%RH;

Temp.: 00FA= 15×16+10 = 250 => Temp. = 250÷10 = 25.0℃

Note： CRC verification code CRC is calculated by the code to arrive, and then pass the CRC and the sensor compared; the same, the data is received correctly, or that the data has errors.

二、 **Function code "10":** write multiple registers to sensor

The host can use this function code multiple data stored in the register to the sensor. AM2315 sensors register a single one-byte or 8 bits. Sensors allow you to save a maximum of 10 data registers. Therefore, the host single sensor to save up to 10 registers. More than 10, the sensor will return the corresponding error code.

Host sends write frame format：

START+(I^2C Add+W)+ Function code (0x10)+ Start address + Register number + Save the data +CRC+STOP

Confirm the host to read instructions：

START+(I^2C Add+R)+ Read the sensor data returned +STOP

Sensor response frame format：

Function code（0x10）+ Start Add + Register number +CRC

For example： Should save the address of the host 01, 02 10, 11 of the sensor to register.

Host sends the message format:

Host sends	Bytes	Send	Remarks
Sensor Add	1	0xB8	Sensor I^2C Add （0xB8）+W（0）
Function code	1	0x10	Write multiple registers
Start Add	1	0x10	To write the start address register
Byte length	1	0x02	Save the data word length (2 Byte)
Data 1	1	0x01	Save the data (address: 10)
Data 2	1	0x02	Save the data (address: 11)
CRC Code	2	C092	Host calculated CRC code, low byte first (I^2C addresses not included in the CRC calculation)

Sensor response to the return message format：

Response from machine	Bytes	Send message	Remarks
Function Code	1	0x10	Write multiple registers
Start address	1	0x10	Save the start address
Save the data length	1	0x02	Sensors save data length
CRC Code	2	FC04	Sensors return the CRC calculation, the low byte first ;

◎ **CRC Check**

Host or check code can be used to discriminate the sensor to receive information is correct. As the electronic noise or some other interference, the information during transmission error sometimes occurs, the error check code (CRC) can verify that the host or sensor data transmission process in the communication of information is wrong, wrong data can be discarded (either send or receive), which increases the system's security and efficiency.

I^2C_ModBus protocol of CRC (cycle redundancy code) consists of 2 bytes, or 16-bit binary number. CRC code from the sending device (host), is placed in the rear send a message frame, I^2C address is not included in the CRC calculation. CRC are either sent or received by the low byte first, high byte format after sending.

Receive information (sensors) and then recalculate the information received by the CRC, the CRC is calculated by comparing the receiver to match, if they do not match, then the error. Users need special

Do note that the sensor reading instruction without adding the CRC; write sensor, the CRC must increase; and all the return data are CRC.

◎ **CRC code is calculated**

1. Preset a 16-bit registers as hexadecimal FFFF (that is, all 1); call this register is the CRC register;
2. The first 8-bit binary data (communication of information both the first byte of the frame) and 16-bit CRC register, or the lower 8 bits are different, the results put in the CRC register;
3. The CRC register right one (towards low) with 0 fill the highest place, and check out right after the bit;
4. If out of the bit is 0: Repeat Step 3 (again shifted to the right one); if out of place as 1: CRC register with the polynomial A001 (1010 0000 0000 0001) XOR;
5. Repeat steps 3 and 4, until the right 8 times, so that all of the 8-bit data were processed;
6. Repeat steps 2 through 5, the communication frame to the next byte of information processing;
7. All bytes of the communication of information frames calculated according to the above steps completed, the resulting 16-bit CRC register is the high and low bytes are exchanged;
8. Finally, get the contents of the CRC register is: CRC code.

◎ **CRC code calculation code in C language**

Description: This program calculates the length of * ptr within the first len bytes of the CRC.

```
unsigned short crc16(unsigned char *ptr, unsigned char len)
{
    unsigned short crc=0xFFFF;
    unsigned char i;
    while(len--)
    {
        crc ^=*ptr++;
        for(i=0;i<8;i++)
        {
            if(crc & 0x01)
            {
                crc>>=1;
                crc^=0xA001;
             }else
             {
                crc>>=1;
             }
        }
    }
    return crc;
}
```

7.2.4 I²C Communication Timing

AM2315 sensor I²C communication, although communication is based on standard I²C timing, but necessary according to our protocol and communication timing requirements, in order to accurately read the sensor. Please strictly in accordance with protocol design and timing for reading.

◎ **I^2C Read the complete sequence example**

Figure 13 shows a complete example of the sensor to read and write and read and write special time requirements, in strict accordance with specific time requirements to read and write, or they will not read the sensor or data appear incorrect and so on. Timing diagram of several special needs attention, detailed figure of the time requirements; host communication from start to finish, for a maximum of 3S.

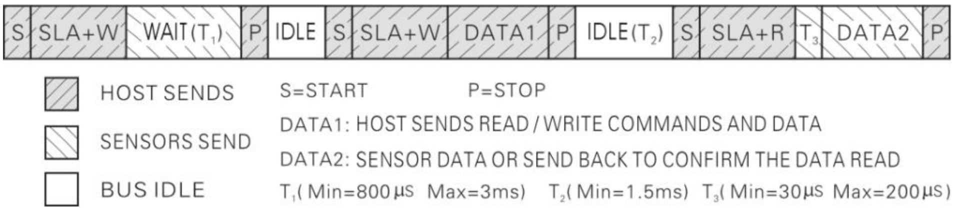


Figure 13: I^2C Sensor reading and writing a complete sample chart

◎ **I^2C Read and write timing decomposition**

Read or write the sensor, the following three steps must be, otherwise it will not communicate or can not read the correct data:

Step one: Wake-up sensor

In order to reduce the humidity sensor self-heating caused the error, the sensor in the non-working state, in a dormant state, so the sensor must wake up to read the sensor, in order to send read and write commands, otherwise the sensor will not respond. It should be noted that, in the wake sensor, sending the I2C address, the sensor will not respond to ACK, but the host must send ACK back the clock to confirm that the ninth SCL clock signal. Wake up the sensor's operating instructions as follows:

Sends a start signal is applied to host the start address, wait for a period of time (to wait for at least 800μ s, the largest 3ms; if the host is a hardware I2C, you do not need to wait, to wait for hardware I2C automatically), then sends a stop signal.

Namely: the initial signal +0 xB8 + wait (800us-3ms) + stop signal timing diagram shown in Figure 14.

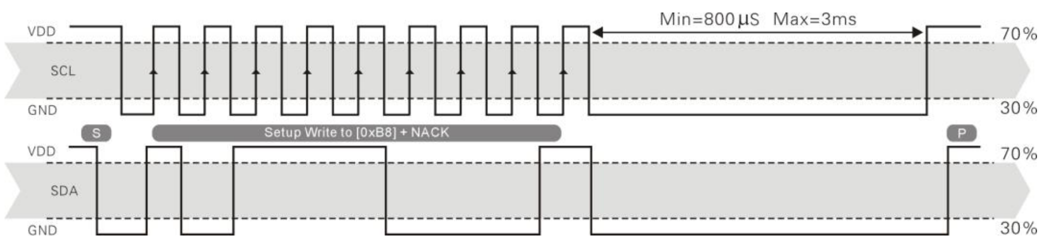


Figure 14: Wake-up sensor

Step Two read commands to send or send written instructions

Wake AM2315 sensors, can be read in full accordance with standard I^2C timing, the maximum speed supported 100Kb / s. Read temperature and humidity sample, shown in Figure 15.

Host to send commands to: START +0 xB8 (SLA) +0 x03 (function code) +0 x00 (start address) +0 x04 (register length) + STOP

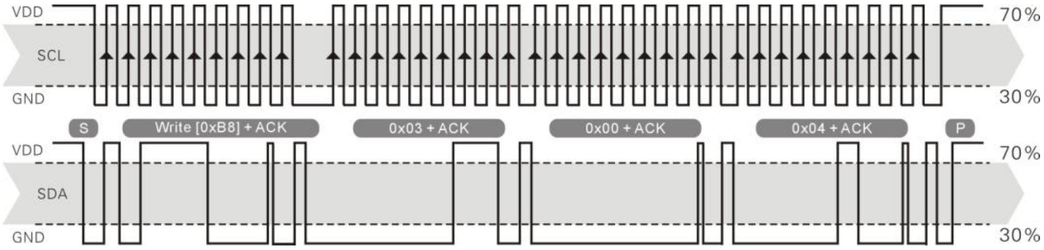


Figure 15：Sends a read command example temperature and humidity

Step three data read back or confirmation signal

Send read / write command, the host must wait at least 1.5ms, and then send a read sequence, to read back the data example shown in Figure 16; to note is that you read the data, finished his I2C address is required wait at least 30μ s before sending over the next serial clock, the read data, or communication error will occur.

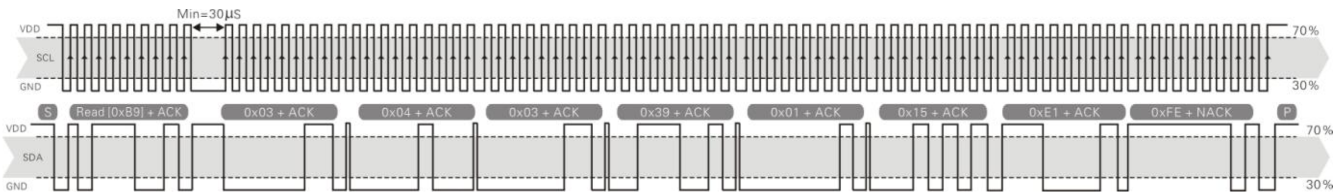


Figure 16: Example values of temperature and humidity reading

Data is read back to the host：

0x03(Function code)+0x04(Data length)+0x03(High humidity)+0x39(Low humidity)+ 0x01(High temperature)+0x15(Low temperature)+0xE1(CRC Low byte checksum)+ 0xFE(CRC Checksum high byte)；

So： 0339H = 3×256+3×16+9 = 825 => Humidity = 825÷10=82.5%RH;

0115H = 1×256+1×16+5 = 277 => Temperature = 277÷10=27.7℃

Through the above three steps to complete all the registers of the sensor reading and writing operations (users can write registers, only five, namely, the status register, four user registers at the same time, the status register, can be written separately, otherwise an error); users in the design, follow these three steps must be fully read and write.

Sensors send the data, trigger a temperature and humidity measurement; measurement is completed, record temperature and humidity values, then a communication completed, the sensor automatically into hibernation; so long as the sensor does not read, read the second consecutive sensors, back to the second reading of the temperature and humidity for the latest value (minimum interval of continuous reading 2S).

7.2.5 Peripherals read the flow chart

AM2315 schematic diagram I2C sensor reading is shown in Figure 17, while our company also provides sample code to read C51, customers need to download, please visit our website (www.aosong.com) associated download, this manual does not provide code instructions.

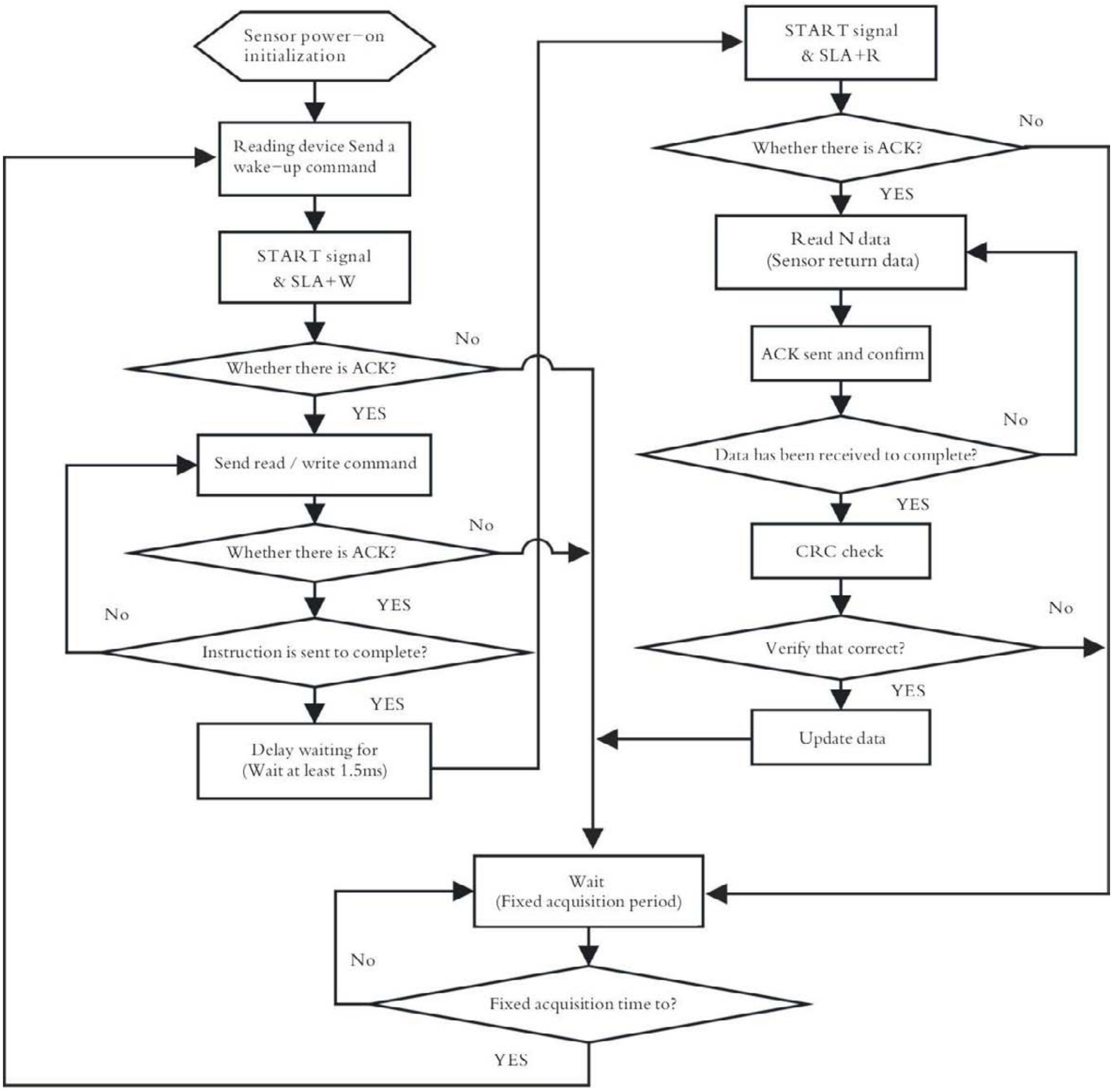


Figure 17: Sensor with I2C read flow chart

Table 1: I^{2}C_MODBUS communication protocol summary table

Read the bus Description: I2C address is 0xB8; access a maximum of 10 registers;

Read the bus communication time for a maximum of a 3 S。

Each returned sensor data plus the CRC, the user can choose not to read the CRC。

Sensor reading frame format: Host frame format: (SLA+W)+ Function code（0x03）+ Start Add + Register number

From the machine frame format: Function code（0x03）+ Data length + Return data +CRC

Write frame format sensor: Host frame format: (SLA+W)+ Function code（0x10）+ Start Add + Register number + Save the data +CRC

From the machine frame format: Function code（0x03）+ Start Add + Register number +CRC

AM2315 sensor registers list:

Register information	Add	Register information	Add	Register information	Add	Register information	Add
High RH	0x00	Model high	0x08	Users register 1 high	0x10	Retention	0x18
Low RH	0x01	Model low	0x09	Users register 1 low	0x11	Retention	0x19
High temp.	0x02	Version number	0x0A	Users register 2 high	0x12	Retention	0x1A
Low temp.	0x03	ID(24−31) Bit	0x0B	Users register 2 low	0x13	Retention	0x1B
Retention	0x04	ID(16−23) Bit	0x0C	Retention	0x14	Retention	0x1C
Retention	0x05	ID(8 − 15) Bit	0x0D	Retention	0x15	Retention	0x1D
Retention	0x06	ID(0 − 7) Bit	0x0E	Retention	0x16	Retention	0x1E
Retention	0x07	Status Register	0x0F	Retention	0x17	Retention	0x1F

Status Register Definition: Bit7−Bit0 bit reserved;

Temperature format: temperature highest (Bit15) equal to 1 indicates a negative temperature, the temperature highest (Bit15) is equal to 0 for positive temperature; temperature in addition to the highest bit (Bit14 ~ Bit0) that string out of the temperature sensor. String out of the temperature sensor is 10 times the actual humidity value;

Write sensor: Register is available for users to write (0x0F ~ 0x13); others prohibit write register and status register can only be written separately。

Reading and writing sample:

Function	code	Start Add	Frame data content
Read temperature and humidity	0x03	0x00	Send：(SLA+W)+0x03+0x00+0x04
			Return：0x03+0x04+ High RH + Low RH + High temp. + Low temp. +CRC
Read the temp.	0x03	0x02	Send：(SLA+W)+0x03+0x02+0x02
			Return：0x03+0x02+ High temp. + Low temp. +CRC
Humidity reading	0x03	0x00	Send：(SLA+W)+0x03+0x00+0x02
			Return：0x03+0x02+ Low RH+ Low RH+ CRC
Read device information	0x03	0x08	Send：(SLA+W)+0x03+0x08+0x07
			Return：0x03+0x07+ Model (16)+ Version (8)+ ID(32)+CRC
Write Status Register	0x10	0x0F	Send：(SLA+W)+0x10+0x0F+0x01+0x01+0xF4（Low）+0xB7（High） Note: Function code + register start address register number + Save + content + CRC
			Return：0x10+0x0F+0x01+0xB4（Low byte）+0x35（High byte） Note：Function code + register start address + register number + CRC
Write user registers 1	0x10	0x10	Send：(SLA+W)+0x10+0x10+0x02+0x01+0x02+0xC0+0x92
			Return：0x10+0x10+0x02+0xFC+0x04

Note: SLA = I2C address 0xB8. Table for the parity bit CRC, CRC for the 16-bit, low byte first, high byte in the post. Return error code: 0x80: does not support the function code 0x81: reading an illegal address 0x82: write data beyond the scope of 0x83: CRC checksum error 0x84: disable writes.

8、the license agreement

Without the copyright holder's prior written permission in any form or by any means, electronic or mechanical (including photocopying), any part of this manual may be reproduced, nor may its contents be communicated to third parties. The contents are subject to change without notice.

The company has a software and third-party ownership, the user only signed a contract or license the software before use.

9、warnings, and personal injury

Do not use this product as safety or emergency stop devices, as well as the product failure could cause personal injury to any other application, unless there are special purpose or authorized use. In the installation, handling, use or maintenance of the product, please consult the product data sheets and application notes. Failure to comply with this proposal, likely to cause death and serious injury. The Company will not assume the resulting personal injury or death of any compensation, and thus exempt from the company's managers and employees and affiliated agents, distributors, etc. that may arise from any claims, including: a variety of costs, compensation costs, legal fees and so on.

10、Quality Assurance

The company's direct purchase of their products to provide a period of 3 months quality assurance (from date of shipment). The company published the product data sheet specifications shall prevail. If the warranty period, the quality of the product proved defective indeed, the company will provide free repair or replacement. User must meet the following conditions:

① This product is found defective within 14 days of written notice to the Company;
② The product shall be paid by mail back to the company;
③ The product should be in the warranty period.

The Company applied only to those who meet the technical requirements of the occasion of the product resulting from defective products responsible. Those companies for their products are used in special applications without any guarantee, warranty or written statement. The company or its products into the product reliability of the circuit does not make any promises.

C.3. PM sensor

Digital universal particle concentration sensor

PMS5003 series data manual

Writer	Zhou Yong	Version	V2.3
Verifier	Zheng Haoxin	Date	2016-06-01

Main characteristics

- Zero false alarm rate
- Real-time response
- Correct data
- Minimum distinguishable particle diameter :0.3 micrometer
- High anti-interference performance because of the patent structure of six sides shielding
- Optional direction of air inlet and outlet in order to adapt the different design

Overview

PMS5003 is a kind of digital and universal particle concentration sensor, which can be used to obtain the number of suspended particles in the air, i.e. the concentration of particles, and output them in the form of digital interface. This sensor can be inserted into variable instruments related to the concentration of suspended particles in the air or other environmental improvement equipments to provide correct concentration data in time.

Working principle

Laser scattering principle is used for such sensor, i.e. produce scattering by using laser to radiate suspending particles in the air, then collect scattering light in a certain degree, and finally obtain the curve of scattering light change with time. In the end, equivalent particle diameter and the number of particles with different diameter per unit volume can be calculated by microprocessor based on MIE theory. Please find the functional diagram of each part of sensor from Figure 1 as follows.

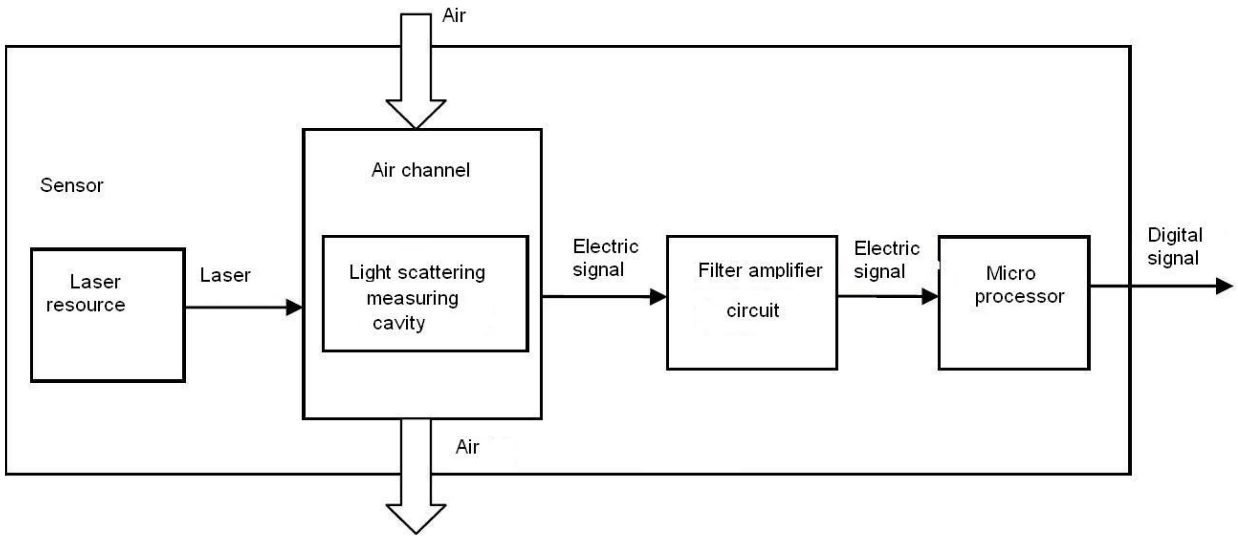


Figure 1 Functional block diagram of sensor

Technical Index

Parameter	Index	unit
Range of measurement	0.3~1.0；1.0~2.5；2.5~10	Micrometer（μ m）
Counting Efficiency	50%@0.3μ m　98%@>=0.5μ m	
Effective Range（PM2.5 standard）	0~500	μ g/m³
Maximum Range（PM2.5 standard）*	≥1000	μ g/m³
Resolution	1	μ g/m³
Maximum Consistency Error (PM2.5 standard data)*	±10%@100~500μ g/m³ ±10μ g/m³ @0~100μ g/m³	
Standard Volume	0.1	Litre（L）
Single Response Time	＜1	Second（s）
Total Response Time	≤10	Second（s）
DC Power Supply	Typ:5.0　Min:4.5　Max: 5.5	Volt（V）
Active Current	≤100	Milliampere（mA）
Standby Current	≤200	Microampere（μ A）
Interface Level	L <0.8 @3.3 H >2.7@3.3	Volt（V）
Working Temperature Range	-10~+60	℃
Working Humidity Range	0~99%	
Storage Temperature Range	-40~+80	℃
MTTF	≥3	Year（Y）
Physical Size	50×38×21	Millimeter（mm）

Note 1: Maximum range means that the highest output value of the PM2.5 standard data is not less than 1000.
Note 2:“PM2.5 standard data” is the “data2” in the appendix.

Pin Definition

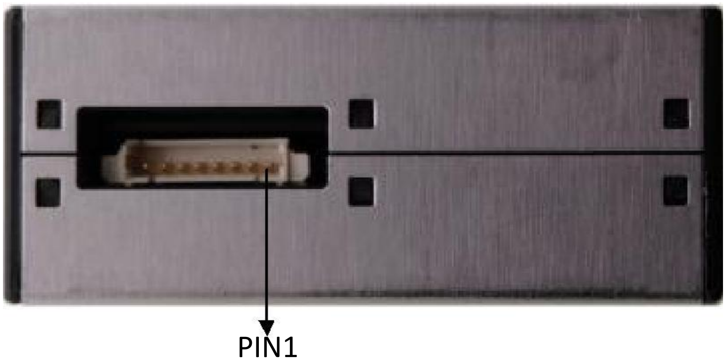


Figure 2 Connector Definition

PIN1	VCC	Positive power 5V
PIN2	GND	Negative power
PIN3	SET	Set pin /TTL level@3.3V，high level or suspending is normal working status, while low level is sleeping mode.
PIN4	RX	Serial port receiving pin/TTL level@3.3V
PIN5	TX	Serial port sending pin/TTL level@3.3V
PIN6	RESET	Module reset signal /TTL level@3.3V，low reset.
PIN7/8	NC	

Output result

Mainly output as the quality and number of each particles with different size per unit volume, the unit volume of particle number is 0.1L and the unit of mass concentration is μ g/m³.

There are two options for digital output: passive and active. Default mode is active after power up. In this mode sensor would send serial data to the host automatically .The active mode is divided into two sub-modes: stable mode and fast mode. If the concentration change is small the sensor would run at stable mode with the real interval of 2.3s.And if the change is big the sensor would be changed to fast mode automatically with the interval of 200~800ms, the higher of the concentration, the shorter of the interval.

Typical Circuit

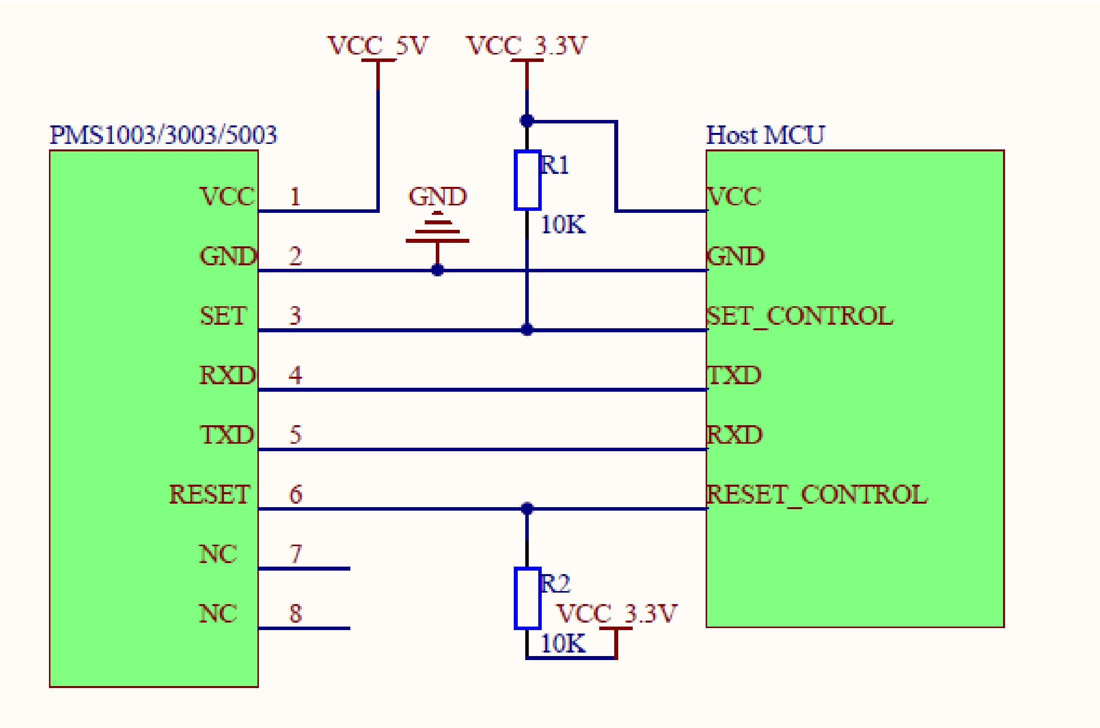


Figure 3 Typical Circuit

Typical Output Characteristic

Definition of axis Y: PM2.5 concentration , unit: μ g/m³

Definition of axis X: number of samples, unit: time

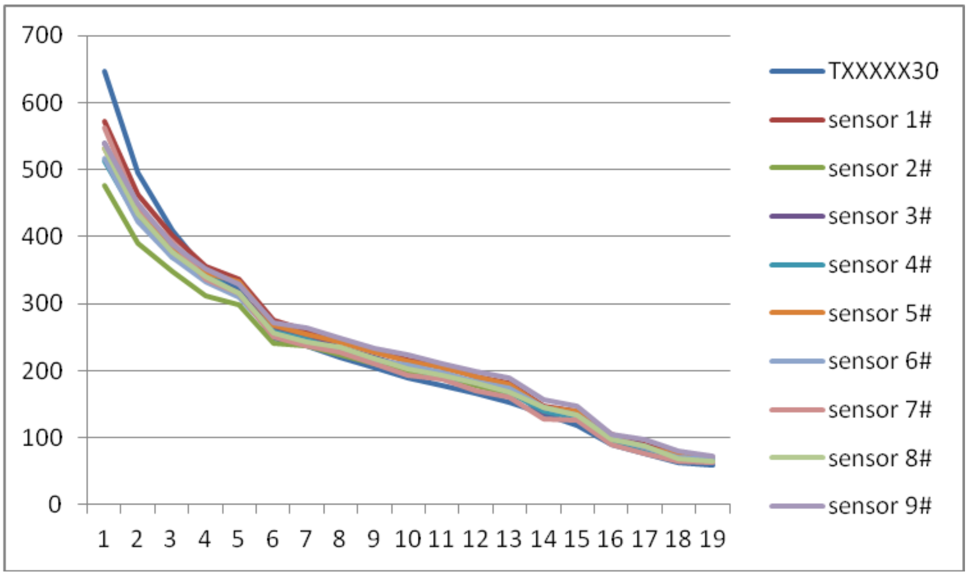


Figure 4-1 Consistency at 20℃

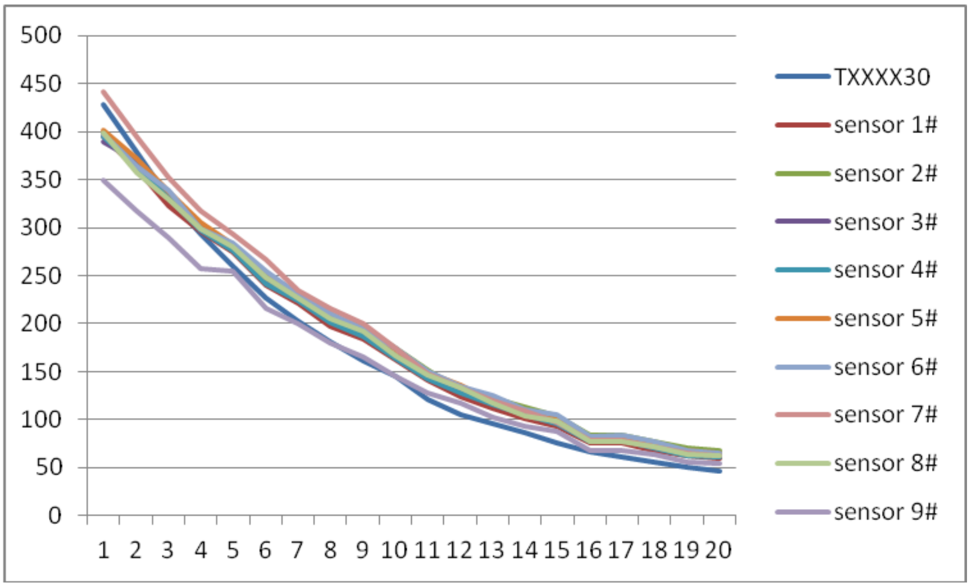


Figure 4-2 Consistency at 43℃

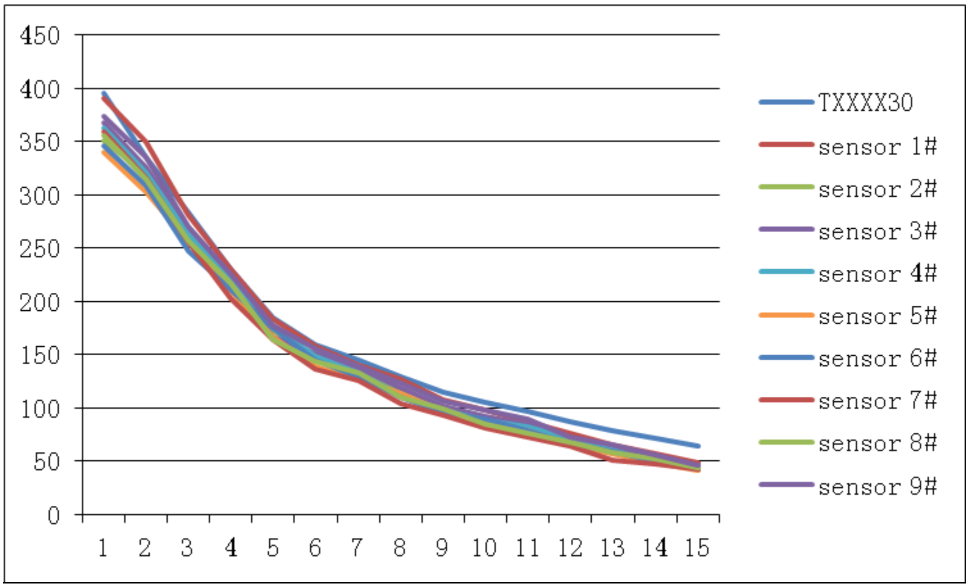


Figure 4-3 Consistency at -5℃

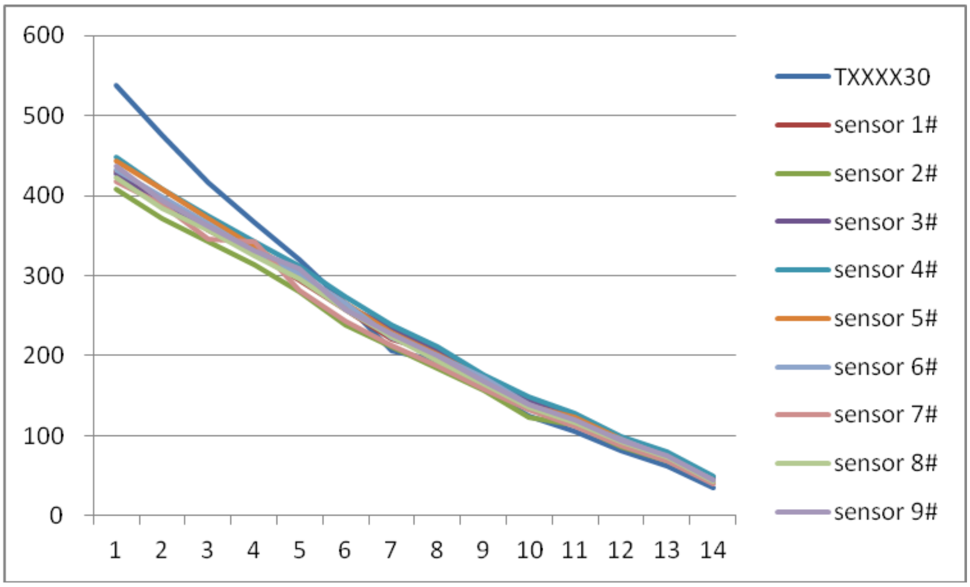


Figure 4-4 Consistency after 30 days' running

Relationship of Temperature and Consistency

Definition of axis Y: Maximum Error Modulus(%)
Definition of axis X: Temperature(℃)

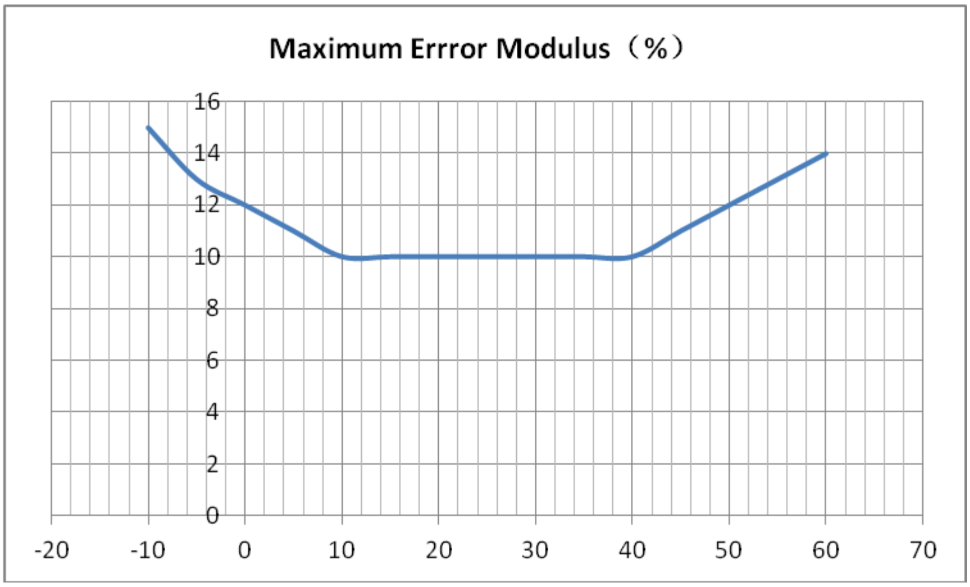


Figure 5 Consistency Vs Temperature

Endurance Characteristics

No	Item	Test Method	Characteristics	n C
1	Long Running	1. 10 ㎡ closed Lab,，20~25℃，humidity 30%~70%，particle generator and air cleaner 2. DC 5V power supply 3. Check consistency after 720 hours' running	10 samples during 0~500μ g/m³ 0~100μ g/m³ Maximum Error≤ ±15μ g/m³ 100~500μ g/m³ Maximum Error≤ ±15% FAN does not screeched	n=30 C=0
2	High Temperature Operation	1. 10 ㎡ constant temperature Lab 2. 43℃，humidity 70%， 3. particle generator and air cleaner 4. DC 5V power supply 5. Check consistency		n=10 C=0
3	Cold Operation	1. 10 ㎡ constant temperature Lab 2. -5℃，humidity 30%， 3. particle generator and air cleaner 4. DC 5V power supply 5. Check consistency		n=10 C=0
4	Vibration	1. 10 ㎡ closed Lab,，20℃，humidity 50%，particle generator and air cleaner 2. DC 5V power supply and check consistency 3. Frequency：50Hz。 4. acceleration：9.8/ S² 。 5. Direction：X、Y、Z 6. Vibration Amplitude：±2mm。 7. Time：X、Y、Z –way, Per 1 hour		n=5 C=0
5	High Temperature and Humidity Storage	1. Constant temperature cabinet 2. 70℃，humidity 90%~95， 3. Check consistency after 500 hours' storage	10 samples during 0~500μ g/m³ 0~100μ g/m³ Maximum Error≤ ±10μ g/m³ 100~500μ g/m³ Maximum Error≤ ±10%	n=10 C=0
6	Cold Storage	1. Constant temperature cabinet 2. -30℃，humidity 90%~95， 3. Check consistency after 500 hours' storage		n=10 C=0
7	Variation of	4. 10 ㎡ closed Lab,，20℃，humidity		n=5

	Power Supply	50%， particle generator and air cleaner 5. Power varies as the cycles of 4.5V to 5.5V ,then 5.5V to 4.5V with the pace of 0.1V/min for 2 hours. 6. Check consistency during Variation	FAN does not screeched	C=0
8	Power On-Off Cycle	1. 10 ㎡ closed Lab,，20℃，humidity 50%， particle generator and air cleaner 2. DC 5V power supply，keep On-Off frequency 0.5Hz for 72 hours and check consistency		n=10 C=0
9	Sleep Set On-Off Cycle	1. 10 ㎡ closed Lab,，20℃，humidity 50%， particle generator and air cleaner 2. DC 5V power supply， keep Sleep Set Pin High-Low frequency 0.5Hz for 72 hours and check consistency		n=10 C=0
10	Laser On-Off Cycle	1. 10 ㎡ closed Lab,，20℃，humidity 50%， particle generator and air cleaner 2. keep laser On-Off frequency 50Hz for 240 hours and check consistency		n=10 C=0
11	Salt Spray	5% industrial salt water, hydrolysis spray 100 hours, clean with purified water and store for 48 hours	No rust and discoloration of metal parts	n=1 C=0

Circuit Attentions

1) DC 5V power supply is needed because the FAN should be driven by 5V. But the high level of data pin is 3.3V. Level conversion unit should be used if the power of host MCU is 5V.
2) The SET and RESET pins are pulled up inside so they should not be connected if without usage.
3) PIN7 and PIN8 should not be connected.
4) Stable data should be got at least 30 seconds after the sensor wakeup from the sleep mode because of the fan's performance.

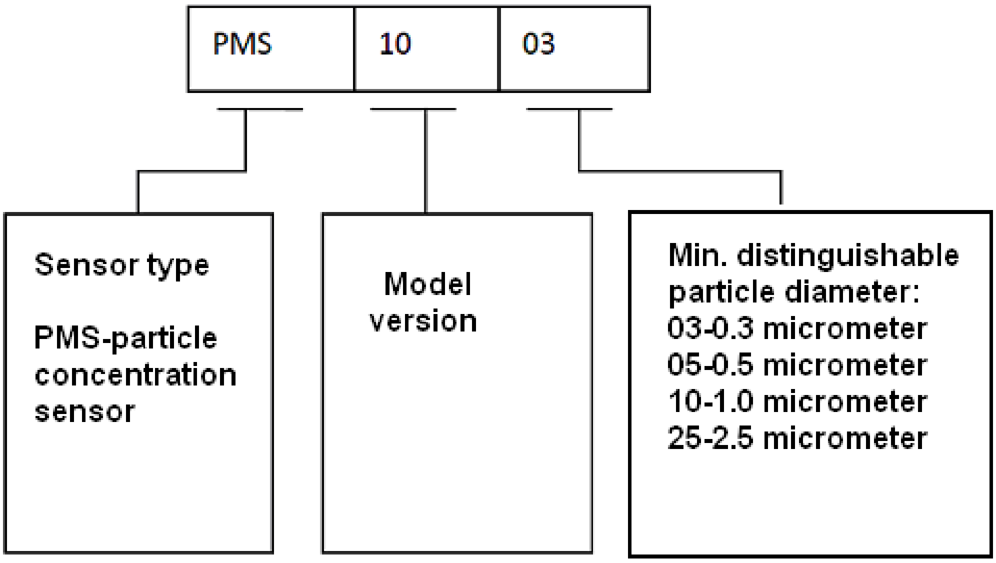
Installation Attentions

1) Metal shell is connected to the GND so be careful not to let it shorted with the other parts of circuit except GND.
2) The best way of install is making the plane of inset and outset closely to the plane of the host. Or some shield should be placed between inset and outset in order to prevent the air flow from inner loop.
3) The blowhole in the shell of the host should not be smaller than the inset.
4) The sensor should not be installed in the air flow way of the air cleaner or should be shielded by some structure.
5) The sensor should be installed at least 20cm higher than the grand in order to prevent it from blocking by the floc dust.
6) Do not break up the sensor.
7) M2 self-tapping strew should be used to fix the sensor but it should not be deeper than 5mm into the sensor.

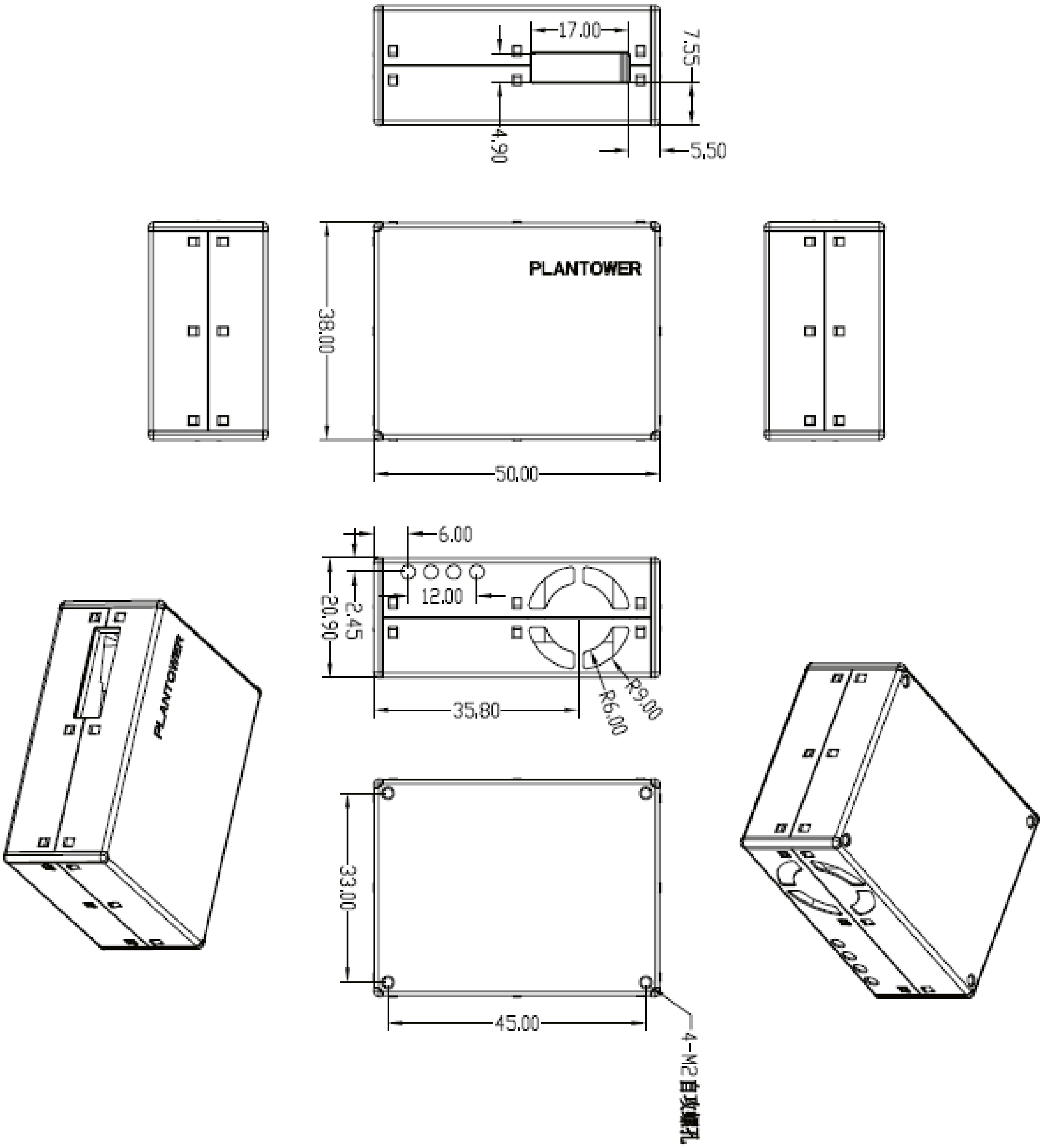
Other Attentions

1) Only the consistency of all the PM sensors of PLANTOWER is promised and ensured. And the sensor should not be checked with any third party equipment.
2) The sensor is usually used in the common indoor environment. So some protection must be added if using in the conditions as followed:
   a) The time of concentration ≥300μ g/m³ is longer than 50% of the whole year or concentration≥500μ g/m³ is longer than20% of the whole year.
   b) Kitchen
   c) Water mist condition such as bathroom or hot spring.
   d) outdoor

Part Number Definition



Physical Size (mm)



Appendix I：PMS5003 transport protocol-Active Mode

Default baud rate：9600bps Check bit：None Stop bit：1 bit

32 Bytes

Start character 1	0x42 (Fixed)	
Start character2	0x4d (Fixed)	
Frame length high 8 bits	……	Frame length=2x13+2(data+check bytes)
Frame length low 8 bits	……	
Data 1 high 8 bits	……	Data1 refers to PM1.0 concentration unit μ g/m3（CF=1，standard particle）*
Data 1 low 8 bits	……	
Data2 high 8 bits	……	Data2 refers to PM2.5 concentration unit μ g/m3（CF=1，standard particle）
Data2 low 8 bits	……	
Data3 high 8 bits	……	Data3 refers to PM10 concentration unit μ g/m3（CF=1，standard particle）
Data3 low 8 bits	……	
Data4 high 8 bits	……	Data4 refers to PM1.0 concentration unit * μ g/m3（under atmospheric environment）
Data4 low 8 bits	……	
Data5 high 8 bits	……	Data 5 refers to PM2.5 concentration unit μ g/m3（under atmospheric environment）
Data5 low 8 bits	……	
Data6 high 8 bits	…….	Data 6 refers to concentration unit (under atmospheric environment) μ g/m3
Data6 low 8 bits	……	
Data7 high 8 bits	……	Data7 indicates the number of particles with diameter beyond 0.3 um in 0.1 L of air.
Data7 low 8 bits	……	
Data8 high 8 bits	……	Data 8 indicates the number of particles with diameter beyond 0.5 um in 0.1 L of air.
Data8 low 8 bits	……	
Data9 high 8 bits	……	Data 9 indicates the number of particles with diameter beyond 1.0 um in 0.1 L of air.
Data9 low 8 bits	……	

Data10 high 8 bits	……	Data10 indicates the number of particles with diameter beyond 2.5 um in 0.1 L of air.
Data10 low 8 bits	……	
Data11 high 8 bits	……	Data11 indicates the number of particles with diameter beyond 5.0 um in 0.1 L of air.
Data11 low 8 bits	……	
Data12 high 8 bits	……	Data12 indicates the number of particles with diameter beyond 10 um in 0.1 L of air.
Data12 low 8 bits	……	
Data13 high 8 bits	……	Data13 Reserved
Data13 low 8 bits	……	
Data and check high 8 bits	……	Check code=Start character 1+ Start character 2+……..+data 13 Low 8 bits
Data and check low 8 bits	……	

Note: CF=1 should be used in the factory environment
.

Appendix II：PMS5003 transport protocol-Passive Mode

Default baud rate：9600bps   Check bit：None   Stop bit：1 bit

Host Protocol

Start Byte 1	Start Byte 2	Command	Data 1	Data 2	Verify Byte 1	Verify Byte 2
0x42	0x4d	CMD	DATAH	DATAL	LRCH	LRCL

1. Command Definition

CMD	DATAH	DATAL	说明
0xe2	X	X	Read in passive mode
0xe1	X	00H-passive 01H-active	Change mode
0xe4	X	00H-sleep 01H-wakeup	Sleep set

2. Answer
   0xe2: 32 bytes , same as appendix I
3. Verify Bytes :
   Add of all the bytes except verify bytes.

C.4. Noise sensor

CUI DEVICES

MODEL: CMA-4544PF-W | **DESCRIPTION:** ELECTRET CONDENSER MICROPHONE

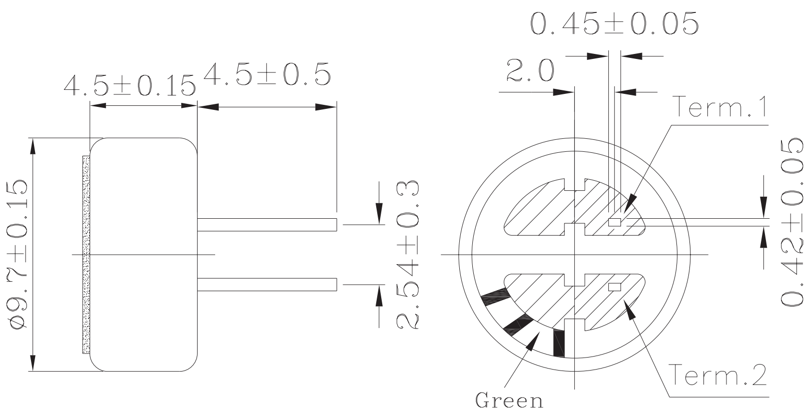
SPECIFICATIONS

parameter	conditions/description	min	typ	max	units
directivity	omnidirectional				
sensitivity (S)	f = 1 kHz, 1 Pa, 0 dB = 1 V/1 Pa	-46	-44	-42	dB
operating voltage			3	10	Vdc
output impedance (Zout)	f = 1 kHz, 1 Pa		2.2		KΩ
sensitivity reduction (ΔS-Vs)	f = 1 kHz, 1 Pa, Vs = 3.0 to 2.0 Vdc		-3		dB
frequency (f)		20		20,000	Hz
current consumption (IDSS)	Vs = 3.0 Vdc, RL = 2.2 KΩ			0.5	mA
signal to noise ratio (S/N)	f = 1 kHz, 1 Pa, A-weighted		60		dBA
operating temperature		-20		70	°C
storage temperature		-20		70	°C
dimension	ø9.7 x 4.5 mm				
weight				0.8	g
material	Al				
terminal	pin type (hand soldering only)				
RoHS	yes				

Note: We use the "Pascal (Pa)" indication of sensitivity as per the recomendation of I.E.C. (International Electrotechnical Commission). The sensitivity of "Pa" will increase 20dB compared to the "ubar" indication. Example: -60dB (0dB = 1V/ubar) = -40dB (1V/Pa)

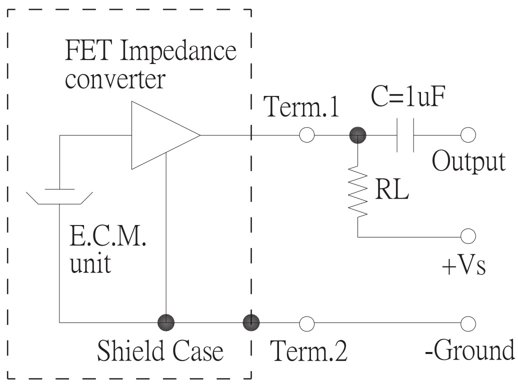
MECHANICAL DRAWING

unit: mm



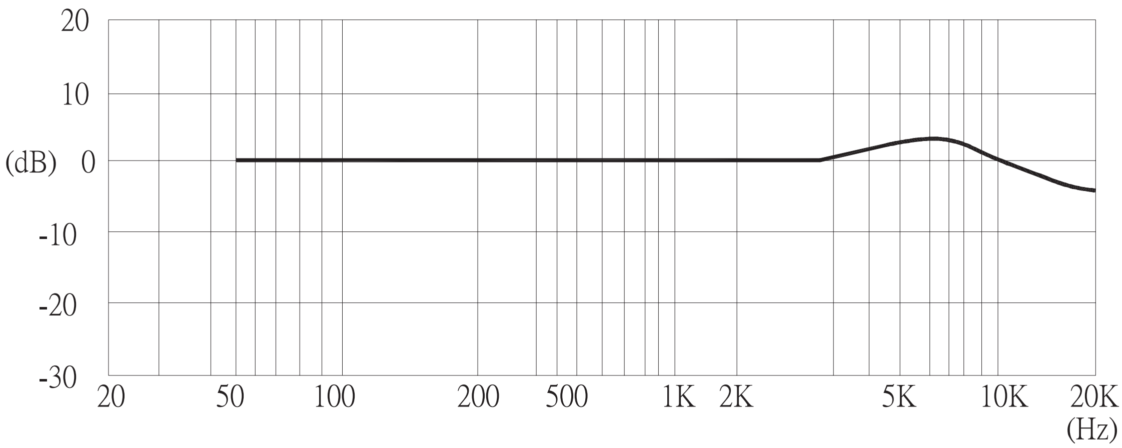
MEASUREMENT CIRCUIT

RL = 2.2 KΩ



Schematic Diagram

FREQUENCY RESPONSE CURVE

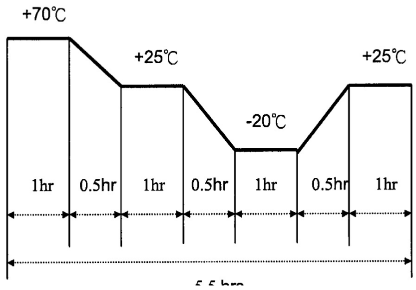


1.000V/PA

MECHANICAL CHARACTERISTICS

item	test condition	evaluation standard
soldering heat resistance	Soldering iron of +270 ±5°C should be placed on the terminal for 2 ±0.5 seconds.	No interference in operation.
terminal mechanical strength	Apply to the terminal 4.9 N (0.5 kg) for 30 seconds	No damage or cutting off.
vibration test	The part should be measured after a vibration amplitude of 1.5 mm with 10~55 Hz band of vibration frequency to each of the 3 perpendicular directions for 2 hours.	After any tests, the sensitivity should be within ±3 dB of the initial sensitivity.
drop test	The part without packaging is subjected to 3 drops on each axis from the height of 1 m onto a 20 mm thick wooden board.	

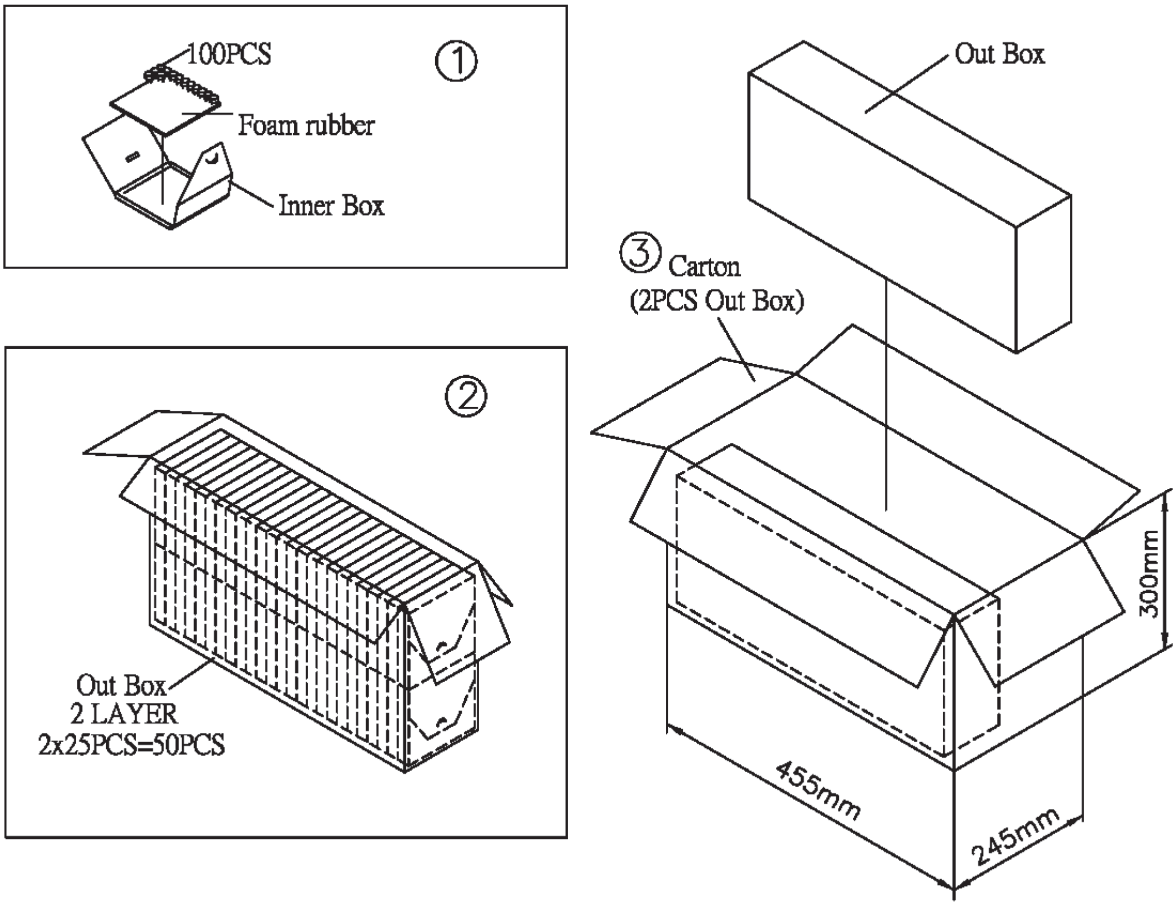
ENVIRONMENT TEST

item	test condition	evaluation standard
high temperature test	After being placed in a chamber at +70°C for 72 hours.	After any tests and 6 hours of conditioning at +25°C, the sensitivity should be within ±3 dB of the initial sensitivity.
low temperature test	After being placed in a chamber at -20°C for 72 hours.	
thermal shock	After being placed in a chamber at +40°C and 90 ±5% RH for 240 hours.	
temperature cycle test	The part will be subjected to 10 cycles. One cycle will consist of: 	

TEST CONDITIONS

standard test conditions	a) Temperature: +5 ~ +35°C	b) Humidity: 45 ~ 85%	c) Pressure: 860 ~ 1060 mbar
judgement test conditions	a) Temperature: +25 ±2°C	b) Humidity: 60 ~ 70%	c) Pressure: 860 ~ 1060 mbar

PACKAGING



Inner Box	100mmx100mmx15mm	100PCSx1=100PCS
Out Box	435mmx120mmx280mm	100PCSx50=5,000PCS
Carton Box	455mmx245mmx300mm	5,000PCSx2=10,000PCS

REVISION HISTORY

rev.	description	date
1.0	initial release	06/01/2008
1.01	new template applied, updated drawing	09/24/2013
1.02	brand update	01/17/2020

The revision history provided is for informational purposes only and is believed to be accurate.

Bibliography

[1] A. Smets, K. Jäger, O. Isabella, R. Van Swaaij, and M. Zeman, *Solar Energy, The Physics and Engineering of Photovoltaic Conversion Technologies and Systens*. Cambridge, England: UIT Cambridge, 2015.

[2] J. Schube, L. Tutsch, T. Fellmeth, F. Feldmann, M. Weil, A. Harter, J.-I. Polzin, M. Bivour, R. Keding, and S. W. Glunz, "Intense pulsed light in back end processing of solar cells with passivating contacts based on amorphous or polycrystalline silicon layers," *Solar Energy Materials and Solar Cells*, vol. 216, p. 110711, Oct 2020. https://www.sciencedirect.com/science/article/pii/S092702482030310X.

[3] P. Wang, G. Li, M. Wang, H. Li, J. Zheng, L. Yang, Y. Chen, D. Li, and L. Lu, "Numerical study of mono-crystalline silicon solar cells with passivated emitter and rear contact configuration for the efficiency beyond 24% based on mass production technology," *Journal of Semiconductors*, vol. 41, p. 062701, jun 2020. https://doi.org/10.1088/1674-4926/41/6/062701.

[4] K. L. Chopra, P. D. Paulson, and V. Dutta, "Thin-film solar cells: an overview," *Progress in Photovoltaics: Research and Applications*, vol. 12, no. 2□3, pp. 69–92, 2004. https://doi.org/10.1002/pip.541.

[5] T. Ablekim, J. N. Duenow, X. Zheng, H. Moutinho, J. Moseley, C. L. Perkins, S. W. Johnston, P. O'Keefe, E. Colegrove, D. S. Albin, M. O. Reese, and W. K. Metzger, "Thin-film solar cells with 19% efficiency by thermal evaporation of cdse and cdte," *ACS Energy Letters*, vol. 5, pp. 892–896, Mar 2020. https://doi.org/10.1021/acsenergylett.9b02836.

[6] A. M. Bagher, M. M. A. Vahid, and M. Mohsen., "Types of solar cells and application," *American Journal of Optics and Photonics*, vol. 3, pp. 94–113, 2015. 10.11648/j.ajop.20150305.17.

[7] E. Suryanto Hasyim, S. R. Wenham, and M. A. Green, "Shadow tolerance of modules incorporating integral bypass diode solar cells," *Solar Cells*, vol. 19, pp. 109–122, Dec 1986. https://www.sciencedirect.com/science/article/pii/0379678786900360.

[8] W. Maier, A. Maier, and V. Maier, "Ae solar alternative energy," 2019. https://ae-solar.com/wp-content/uploads/2018/10/AE_Solar_Presentation_ENG.pdf.

[9] M. A. Green, E. Gauja, and W. Withayachamnankul, "Silicon solar cells with integral bypass diodes," *Solar Cells*, vol. 3, pp. 233–244, May 1981. https://www.sciencedirect.com/science/article/pii/0379678781900053.

[10] R. Ramaprabha and B. L. Mathur, "A comprehensive review and analysis of solar photovoltaic array configurations under partial shaded conditions," 2012. https://doi.org/10.1155/2012/120214.

[11] S. R. Pendem and S. Mikkili, "Modeling, simulation, and performance analysis of pv array configurations (series, series-parallel, bridge-linked, and honey-comb) to harvest maximum power under various partial shading conditions," *International Journal of Green Energy*, vol. 15, no. 13, pp. 795–812, 2018. 10.1080/15435075.2018.1529577.

[12] A. Thomson, M. Ernst, I. Haedrich, and J. Qian, "Impact of pv module configuration on energy yield under realistic conditions," *Optical and Quantum Electronics*, vol. 49, p. 82, 01 2017. 10.1007/s11082-017-0903-0.

[13] J. Qian, A. Thomson, A. Blakers, and M. Ernst, "Comparison of half-cell and full-cell module hotspot-induced temperature by simulation," *IEEE Journal of Photovoltaics*, vol. 8, no. 3, pp. 834–839, 2018. 10.1109/JPHOTOV.2018.2817692.

[14] K. R. McIntosh, R. M. Swanson, and J. E. Cotter, "A simple ray tracer to compute the optical concentration of photovoltaic modules," *Progress in Photovoltaics: Research and Applications*, vol. 14, no. 2, pp. 167–177, 2006. https://doi.org/10.1002/pip.647.

[15] A. Joshi, A. Khan, and A. SP, "Comparison of half cut solar cells with standard solar cells," in *2019 Advances in Science and Engineering Technology International Conferences (ASET)*, pp. 1–3, 2019. 10.1109/ICASET.2019.8714488.

[16] S. Guo, J. P. Singh, I. M. Peters, A. G. Aberle, and T. M. Walsh, "A quantitative analysis of photovoltaic modules using halved cells," *International Journal of Photoenergy*, vol. 2013, Sep 2013. 10.1155/2013/739374.

[17] F. Kaule, M. Pander, M. Turek, M. Grimm, E. Hofmueller, and S. Schoenfelder, "Mechanical damage of half-cell cutting technologies in solar cells and module laminates," *AIP Conference Proceedings*, vol. 1999, no. 1, 2018. 10.1063/1.5049252.

[18] A. Janik-Hannen, *ITRPV - VDMA*. International Technology Roadmap for Photovoltaic, 2021. https://itrpv.vdma.org/en/.

[19] MATLAB, *(R2020a)*. Natick, Massachusetts: The MathWorks Inc., 2020.

[20] Velleman, *Sol3n data sheet*, Februari 2017. https://www.velleman.eu/downloads/6/infosheets/sol3n_datasheet.pdf.

[21] *AKW data sheet*.

[22] *Unknown solar data sheet*.

[23] V. J. Chin, Z. Salam, and K. Ishaque, "Cell modelling and model parameters estimation techniques for photovoltaic simulator application: A review," *Applied Energy*, vol. 154, pp. 500–519, 2015. https://doi.org/10.1016/j.apenergy.2015.05.035.

[24] J. Ramos Hernanz, J. Campayo Martin, I. Zamora Belver, J. Larranaga Lesaka, E. Zulueta Guerrero, and E. Puelles Perez, "Modelling of photovoltaic module," in *International conference on renewable energies and power quality (ICREPQ'10)*, pp. 23–25, 2010.

[25] V. Stornelli, M. Muttillo, T. de Rubeis, and I. Nardi, "A new simplified five-parameter estimation method for single-diode model of photovoltaic panels," *Energies*, vol. 12, Feb 2019. https://www.proquest.com/scholarly-journals/new-simplified-five-parameter-estimation-method/docview/2317092941/se-2?accountid=27026.

[26] LTspice, *(XVII)*. Wilmington, Massachusetts: Analog Devices, 2016.

[27] U. Dutta and P. Chatterjee, "The open circuit voltage in amorphous silicon p-i-n solar cells and its relationship to material, device and dark diode parameters," *Journal of applied physics*, vol. 96, no. 4, pp. 2261–2271, 2004.

[28] P. Trinuruk, C. Sorapipatana, and D. Chenvidhya, "Estimating operating cell temperature of bipv modules in thailand," *Renewable Energy*, vol. 34, no. 11, pp. 2515–2523, 2009. https://doi.org/10.1016/j.renene.2009.02.027.

[29] Y. An, C. Sheng, and X. Li, "Radiative cooling of solar cells: opto-electro-thermal physics and modeling," *Nanoscale*, vol. 11, pp. 17073–17083, 2019. https://doi-org.tudelft.idm.oclc.org/10.1039/C9NR04110A.

[30] J. Hernandez-Perez, J. Carrillo, A. Bassam, M. Flota-Banuelos, and L. Patino-Lopez, "Thermal performance of a discontinuous finned heatsink profile for pv passive cooling," *Applied Thermal Engineering*, vol. 184, p. 116238, 2021. https://doi.org/10.1016/j.applthermaleng.2020.116238.

[31] K. K. Dixit, I. Yadav, G. K. Gupta, and S. Kumar Maurya, "A review on cooling techniques used for photovoltaic panels," in *2020 International Conference on Power Electronics IoT Applications in Renewable Energy and its Control (PARC)*, pp. 360–364, 2020. 10.1109/PARC49193.2020.236626.

[32] L. J. Habeeb, D. G. Mutasher, F. Abid, and M. Abd, "Cooling photovoltaic thermal solar panel by using heat pipe at baghdad climate," *International Journal of Mechanical & Mechatronics Engineering IJMME-IJENS*, vol. 17, no. 6, pp. 171–185, 2018. https://www.researchgate.net/profile/Dheya_Ghanim/publication/329196967_Cooling_Photovoltaic_Thermal_Solar_Panel_by_Using_Heat_Pipe_at_Baghdad_Climate/links/5bfc5d05a6fdcc76e722a904/Cooling-Photovoltaic-Thermal-Solar-Panel-by-Using-Heat-Pipe-at-Baghdad-Climate.pdf.

[33] CUI Devices, *Electret Condenser Microphone*, 2020. https://www.cuidevices.com/product/resource/cma-4544pf-w.pdf.

[34] PLANTOWER, *Digital universal particle concentration sensor*, 2016. Rev. 2.3 https://cdn-shop.adafruit.com/product-files/3686/plantower-pms5003-manual_v2-3.pdf.

[35] AOSONG, *Digital temperatures and humidity sensor*. https://cdn-shop.adafruit.com/datasheets/AM2315.pdf.

[36] Bosch Sensortec, *Digital Pressure Sensor*, 2015. Rev. 1.14 https://cdn-shop.adafruit.com/datasheets/BST-BMP280-DS001-11.pdf.

[37] *Photovoltaic Geographical Information System (PVGIS)*. EU Science Hub - European Commission, 2019. https://ec.europa.eu/jrc/en/pvgis.