UNIVERSITY OF TARTU
Institute of Computer Science
Computer Science Curriculum

Karl Lubja

Energy Harvesting Optimization with Solar Panels

Bachelor's Thesis (9 ECTS)

Supervisor(s): Mohan Liyanage, MSc
Satish Srirama, PhD

Tartu 2017

Energy Harvesting Optimization with Solar Panels

Abstract:

Fossil fuels have been the main source of energy for mankind for several centuries. However, due to the increasingly present danger of global warming and fossil fuels being a finite resource, humans have started looking towards more sustainable and eco-friendly sources of energy. Solar power is one of the most sought out options, but the full potential of the solar panels is yet to have been reached. The aim of this thesis is to provide an overview of the currently existing solar tracking solutions and propose and test a solution based on the Arduino microcontroller, capable of greatly increasing the output of a solar panel. The author developed a prototype consisting of four photoresistors, two servo motors and a solar panel and tested it. The experimental results showed that the proposed system is able to harvest 20% of more energy from the solar panel than the traditional fixed panels.

Keywords:

Arduino, solar panel, photoresistor, servo motor

CERCS: P170 Computer science, numerical analysis, systems, control

Päikesepaneelide energia talletamise optimiseerimine

Lühikokkuvõte:

Inimkond on fossiilkütuseid kasutanud põhilise energiaallikana juba mitu sajandit, kuid fossiilkütused on taastumatu ressurss ning nende kasutamine on äärmiselt keskkonda reostav. See tõttu on inimkond hakanud otsima alternative fossiilkütustele, mis oleksid taastuvad ning keskkonna sõbralikud. Üks põhisuundi on päikeseenergia – tegemist on lõpmatu ja mittereostava energiaallikaga, kuid päikeseenergia täit hetke potentsiaali pole ära kasutatud. Bakalaureusetöö eesmärk on luua nii tarkvaraline kui ka riistvaraline lahendus päikesepaneelide liikumise automatiseerimiseks, et maksimeerida päikeseneenergia talletust. Töö käigus arendati prototüüp, mis koosnes neljast valgustakistist, kahest servo motorist ja ühest päikesepaneelist. Katse tulemustega ilmnes, et päikest jälgiv paneel on keskmiselt võimeline tootma 20% rohkem energiat võrreldes traditisoonilise paigalseisva päikesepaneeliga.

Võtmesõnad:

Arduino, päikesepaneel, valgustakisti, servo motor

CERCS: P170 Arvutiteadus, arvutusmeetodid, süsteemid, juhtimine (automaatjuhtimisteooria) reaalteadused

Table of contents

1. Introduction

Ever since the start of the Industrial Revolution in the 18^{th} century, mankind has been dependent on fossil fuels. Reliance on fossil fuels has always had two major drawbacks. Firstly, usage and production of fossil fuels and materials derived from it are extremely energy hungry and polluting. Secondly, the natural process of fossil fuel formation takes millions of years, meaning there is a finite amount of it. So today's world trend is moving to renewable energy sources more than ever. Among all the potential sustainable energy sources, solar power has become one of the most key resource and will alleviate the problems the usage of fossil fuels pose. The earth receives the average power of 1.4 kW / m^2 [1] from the sun daily and also it is an infinite source of energy transformation and produces zero greenhouse gases in the process.

Solar panels have become more and more common in today's world to provide power for households. Yet most of the commercial solar panels are fitted in a stationary position which lowers their efficiency and output potential. This may suit everyday homes that in addition to solar panels use the power grid for their electricity needs. However, some remote locations, places like national parks and islands, do not have access to the electrical infrastructure and must depend on other means of sources for power, like solar.

Dependency on one source of power, especially solar, means that the volatility of the weather plays a crucial role in the availability of electricity. In order to reduce the risk of having power outages, necessary measures should be taken to maximize the output of solar panels.

The simplest and most efficient way to achieve this goal is to move the solar panels in such a way that the panels are always pointed towards the sun in the most optimal angle. The aim of this thesis is to introduce existing solar tracking solutions, realize and test a solution based on the Arduino microcontroller that is capable of increasing the output of a solar panel by a significant amount.

The rest of the thesis is organized as follows. Chapter 2 describes the background of the technologies and in detail view of the solar tracking. Chapter 3 provides the detail explanation of the proposed system with algorithms and the theories applied. Implementation of the prototype has been described in the Chapter 4 along with the experimental evaluation. Finally, the thesis concludes in Chapter 5 with future research plans.

2. Background

This chapter introduces the Arduino platform, outlines the reasons for the choice of hardware and briefly describes solar power.

2.1 Solar Power

Solar power is the process of converting the energy of light into electricity [2]. There are two types of solar power systems – concentrated solar power systems or direct systems that use the photovoltaic (PV) effect. The former uses lenses and mirrors to direct a larger area of sunlight into a small concentrated beam of light. For this thesis, a direct system is used as the solution created is meant for private usage.

The output of a solar panel is directly related to the angle of incidence θ, shown in Figure 1 below. This means that the higher the angle of incidence θ, the lower is the output of a solar panel [3].

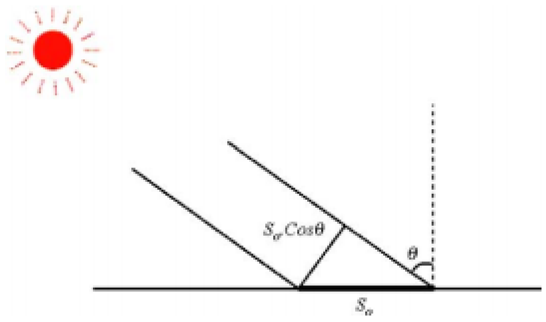

Figure 1. Angle of incidence θ of the solar radiation[3].

2.1.1 Cost

Solar power is thought to be expensive, however, the cost per watt has decreased over 99% from $96 since the mid-1970s [2]. Mass production and improvements in technology have taken the price down to 68 cents per watt taking into account inflation. The lifespan of a PV system is thought to typically span 25 to 40 years. Since direct systems do not use any fuel, most of the cost is made up by capital costs and maintenance costs are estimated to total about 9% in the US.

2.1.2 Solar Tracking

According to the International Energy Agency [4] every square metre of Europe receives an average of 1200 kilowatt-hours of solar energy per year. However, the amount of energy received can be increased by using solar tracking. For example, solar power modules in La Rochelle, France that take advantage of optimal tilt and orientation angles harvest up to 200 $kWh/m^2/y$ more than the normal 1300 $kWh/m^2/y$.

There are two approaches to solar tracking – passive and active. One example of a passive solution is by Marten Lillemäe who implemented a solar tracking system in his bachelor's thesis [2] that used a real-time clock in order to decide whether the machine should be turned on and where the machine should rotate. However, the implementation created in this thesis will use an active solution. Meaning that the sun is tracked via sensors and is more flexible than a real-time clock.

2.2 Existing works

J. Rizk and Y. Chaiko [5] write in their paper "Solar Tracking System: More Efficient Use of Solar Panels" that they built a simple solar tracking system that used a stepper motor, a 9W solar and sensors with the cost being around 100 Australian dollars. In their prototype, a fixed panel provided an average of 3.51W of the maximum 9W. However, using the tracking system they achieved an average of 6.3W. Furthermore, the authors noted in the paper that during the earlier and later hours of the day, they show up to a 400% increase in power compared to the fixed panel. Further examples include the use of a GPS device to calculate the angle of the sun and a stepper motor pinpoint the solar panel [6]. In [7] the authors proposed to use one photo-resister to track the sun and another photo-resister to detect the day and night. However, using only one photo-resister with single axis tracking is not efficient in the Europian countries where the length of the daytime varies largely across the year (from 6 to 18 hrs). Kale et.al [8] presented a Photovoltaic system that adjusting a duty cycle to enhance the charging capacity of a solar power system. The proposed system applied DC to DC boost converter that can control the switching frequency. Moreover, there are no moving parts, instead, the system keeps trying to maintain a constant voltage level across the day, which is more important to charge lead acid batteries.

When compared to the above-mentioned systems, the proposed system consists of four photoresistors which can precisely track the incident angle of the sunlight and also the algorithm and two axis rotation that designed to harvest more solar energy.

2.3 Arduino

The following subsection is referred from the Arduino homepage [9]. Arduino is a completely open-source platform than encompasses hardware and software solutions. Its origins lie at the Ivrea Interaction Design Institute, where in 2005 it was created for students without any experience in electronics and programming. Arduino quickly started to attract a community around its easy usability and affordable price. As the number of contributors grew, Arduino itself started to adapt and evolve for new applications and possibilities – IoT applications, 3D printing, wearable and embedded environments. Alongside its hardware, Arduino's community has developed the Arduino programming language which is based on Wiring and an integrated development environment.

2.3.1 Hardware

Arduino offers over 20 different boards with mainly Atmel processors with CPU speeds ranging from 8 MHz to 400MHz [10]. For this thesis, both Arduino Uno and Mega 2560 were under consideration for their easy setup and a large enough pin count.

Table 1. Comparison of Arduino Uno and Arduino Mega 2560

	Arduino Uno	*Arduino Mega 2560*
Processor	ATmega328P	ATmega2560
CPU speed	16 MHz	16 MHz
Operating/Input Voltage	5V / 7-12V	5V / 7-12V
Analog In/Out	6/0	16/0
Digital IO/PWM	14/6	54/15
SRAM [kB]	32	256

As depicted in the Table 1 above both of the microcontrollers have the same clock speed CPUs, but Mega 2560 has the advantage over Uno because of its larger SRAM size and over twice as many inputs and outputs. This would make the Mega 2560 more suitable for projects that have many inputs and outputs. However, a planned prototype consisting of two servo motors as outputs and four photoresistors as inputs meant that Uno would be sufficient.

2.3.2 Software

Arduino has an open-platform called the Arduino Software, based on Processing [11]. It is available on all the major operating systems – Windows, Mac and Linux. The IDE (Figure 2) supports the languages C and C++ and comes with built-in examples to make the learning process easier for newcomers and beginners [9, 12]. In addition to examples, Arduino IDE contains several built-in libraries and the option of downloading third party ones which provide extra functionality and assist in writing programs [3].

```
/*
  AnalogReadSerial
  Reads an analog input on pin 0, prints the result to the serial monitor.
  Graphical representation is available using serial plotter (Tools > Serial Plotter menu)
  Attach the center pin of a potentiometer to pin A0, and the outside pins to +5V and ground.

  This example code is in the public domain.
*/

// the setup routine runs once when you press reset:
void setup() {
  // initialize serial communication at 9600 bits per second:
  Serial.begin(9600);
}

// the loop routine runs over and over again forever:
void loop() {
  // read the input on analog pin 0:
  int sensorValue = analogRead(A0);
  // print out the value you read:
  Serial.println(sensorValue);
  delay(1);        // delay in between reads for stability
}
```

Figure 2. Arduino IDE Built-In Example

C and C++ are not the only programming languages that can be used to develop on the Arduino. In fact, every language that uses a compiler that transforms the program into binary code can work on the Arduino. However, using another language other than Arduino's own may have less support in the form of libraries and examples. Taking into account the absent libraries, the official language was used for developing on the Arduino.

2.4 Serial Peripheral Interface (SPI)

This subsection [13] is referred from the Arduino web page. Serial Peripheral Interface is a communication protocol used by microcontrollers to communicate with one or even several peripheral devices called slaves. When using SPI there is always one central device that is referred to as the master usually being a microcontroller. Communication is achieved using four lines of connections:

- **MISO** (Master In Slave Out) – Slave's line of communication to send data to the master.
- **MOSI** (Master Out Slave In) – Master's line of communication to send data to the peripheral(s).
- **SCK** (Serial Clock) – Line of communication that is used to synchronize data transmission via clock pulses generated by the master.
- **SS** (Slave Select) – This line is specific for every slave and is used to enable and disable ever device.

2.5 Location

The location for testing the designed system is Tartu in southeastern Estonia. Tartu receives about 950-960 kilowatt hours of energy per square metre per year as shown in Figure 3.

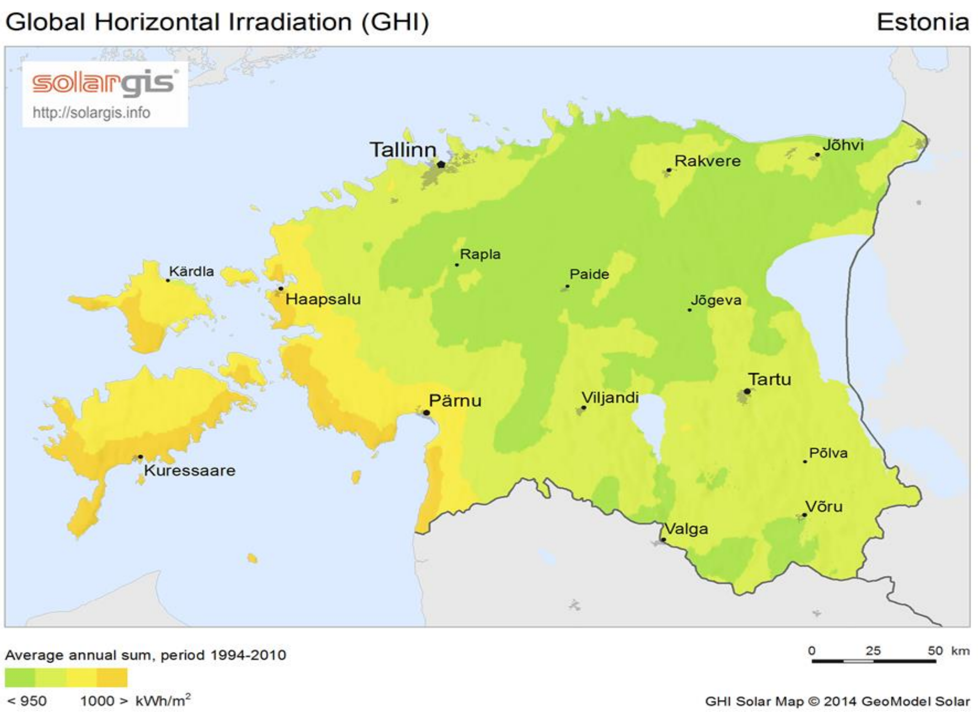

Figure 3. Map of Estonia showing the annual amount of solar energy received [14].

As the International Energy Agency [4] demonstrated in their document "Technology Roadmap Solar Photovoltaic Energy – 2014 Edition" solar arrays in La Rochelle, France achieved a 18.18% increase of received energy per square metre. If the same results are applicable to Tartu, we would expect increased up to ~1123-1135 kilowatt hours of energy per square metre per year. However, in Tartu, expecting such a high gain is unreasonable because of the annual sunshine hours and the geographical conditions of La Rochelle and Tartu.

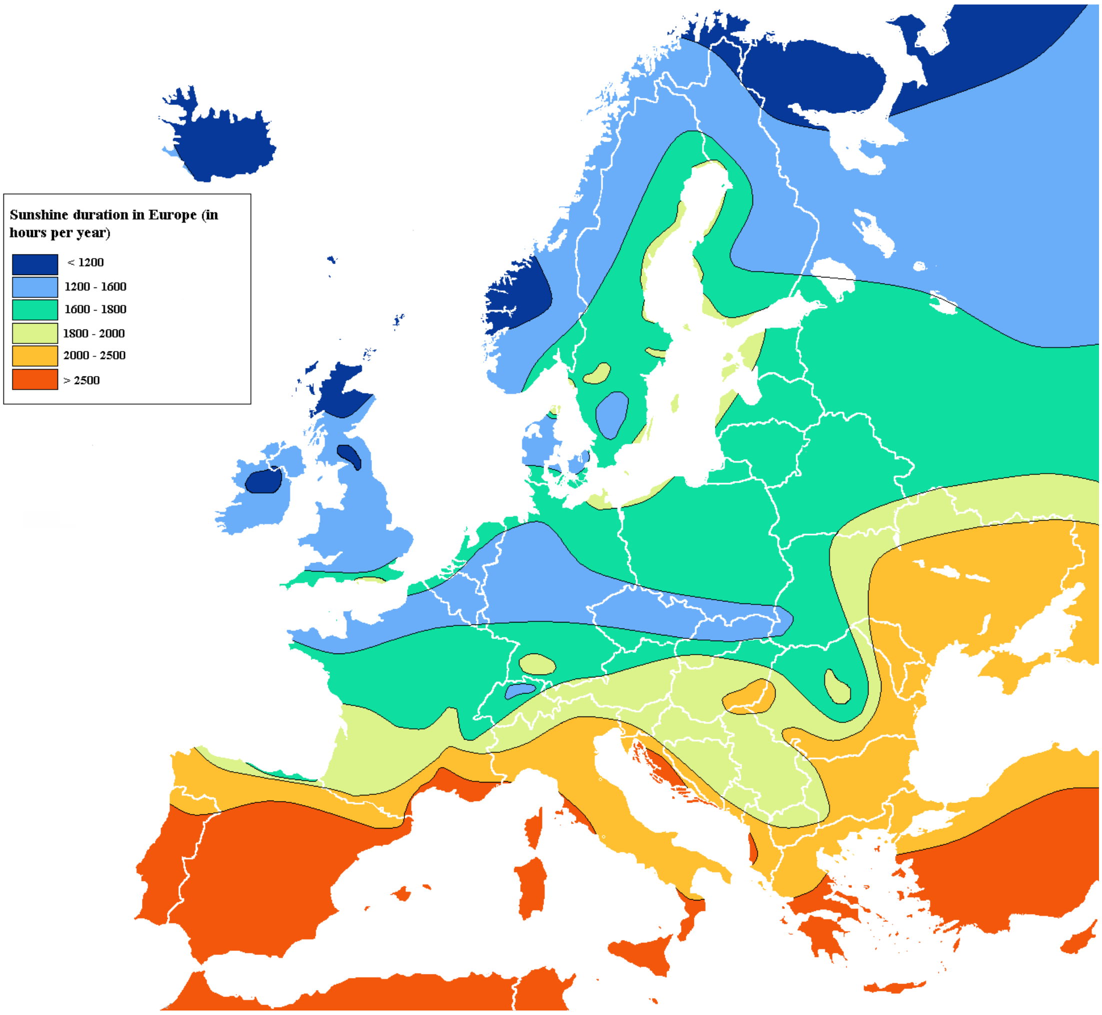

Figure 4. Sunshine duration in Europe (in hours per year) [15].

The gravest contrast between those two locales is the annual sunshine duration in hours per year. As demonstrated in Figure 4, southeastern Estonia receives up to 1800 hours of sunshine per year compared to Southern France which is more than 2500 hours per year.

3. System Design

In this chapter, the author will describe the complete design and various components of the system in detail and outline their functionality.

3.1 Panel movement component design

The proposed system as shown in Figure 5, consists of a controller, 4 light-dependent resistors (LDRs), two servo motors and a solar panel. The LDRs are responsible for detecting the changes in light intensity that the solar panel corners receive and according to the LDRs values that the controller reads, appropriate adjustments are made using the two servo motors connected to the controller.

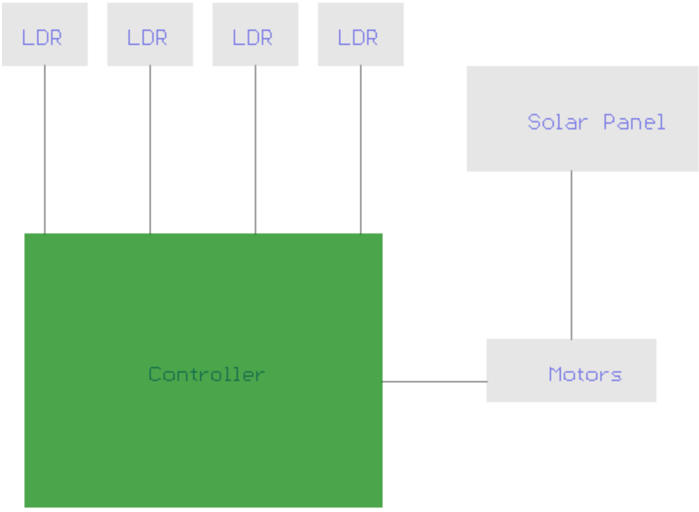

Figure 5. Part of the system that handles pointing of the solar panel

Aforementioned alterations ensure that each of the LDRs receives the same amount of light which means that the solar panel is directed towards the light source, in our case the Sun, in the most optimal and efficient manner.

3.2 Solar panel output component design

Solar panel output is handled by the component depicted in Figure 6 which consists of the solar panel, a current sensor, a controller, SD card module and an inverter with a battery.

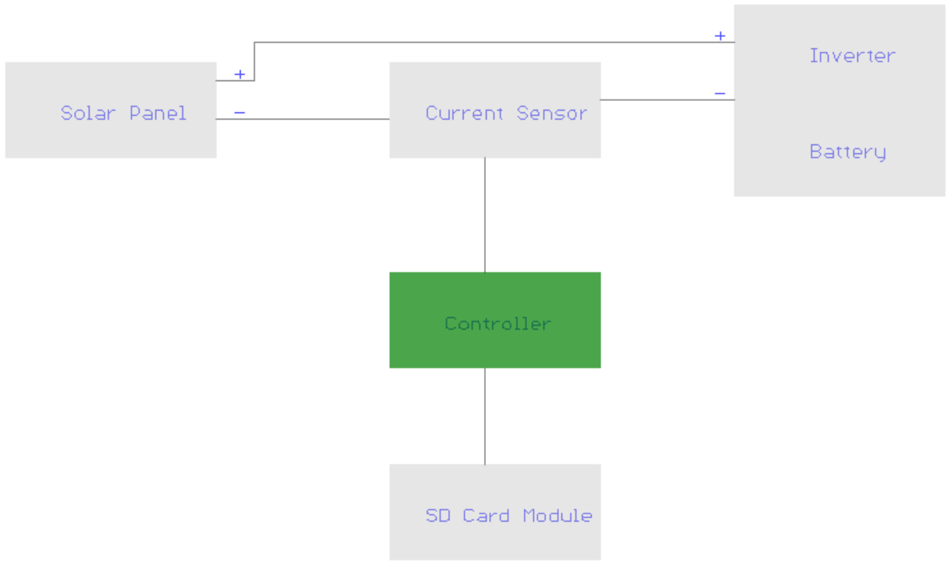

Figure 6. Part of the system that is responsible for reading the output of the solar panel and logging it.

The positive end of the solar panel output goes to the positive end of the battery and from the battery's negative end is connected to the current sensor's inputs and the same applies to the solar panel's negative end. With those connections, a complete circuit is made and thereafter the current sensor is able to read the current flowing through the circuit. The output of the current sensor is handled by a controller. After the input from the current sensor is read, it is logged using the SD card module for future data-analysis. It must be noted that the controller that reads the current sensor input and logs it, is separate from the controller that is responsible for moving the solar panel to avoid potential background noise issues that influence the current sensor readings.

3.3 Complete Hardware Design

The complete hardware design illustrated in Figure 7 that would make up a completely autonomous solar panel system includes the following functionality:

1. Point the solar panel directly at the sun.
2. Log the current generated by the solar panel to a SD card.
3. Charge a battery or a battery system.
4. Feed electricity to the commercial and/or local electrical network.

It must be noted that functionality number 4 was not implemented as this thesis concentrates on maximizing the output of the solar panel.

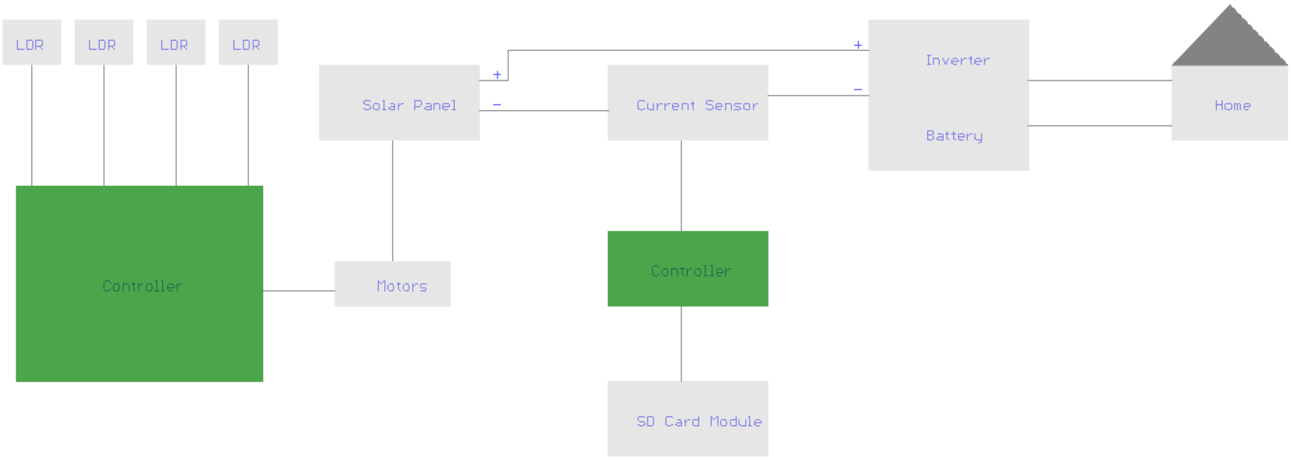

Figure 7. Complete design of the proposed system.

However, an inverter is not implemented in this thesis as inverters are needed to convert direct current into the alternating current so that the output of a solar panel could be fed into domestic use or for an electrical grid. Since the objective of the thesis is to maximize the output of a solar panel, only the battery section was implemented.

3.4 Algorithmic component

The algorithmic component depicted in Figure 8 consists of two separate modules:

- One being responsible for reading the values of the four LDRs and controlling the two servo motors used in the panel movement component.
- The second module is responsible for reading the output current of the solar panel and logging of the current onto the SD card.

The modules are respectively loaded into the Arduino Uno and Arduino Mega ADK.

3.4.1 Solar panel movement algorithm

The algorithm used to control the solar panel movement is depicted in Figure 8. It consists of four main chunks of code:

- Reading of the LDR values.
- Calculating the averages of the solar panel sides.
- Deciding whether and where the solar panel must move.

```
void loop() {
  //Read LDR aka photoresistor values
  upperRightReading = analogRead(upperRight);
  bottomRightReading = analogRead(bottomRight);
  upperLeftReading = analogRead(upperLeft);
  lowerLeftReading = analogRead(lowerLeft);

  //calculate averages for upper and lower
  upperAverage = round((upperRightReading + upperLeftReading)/2);
  lowerAverage = round((lowerLeftReading + bottomRightReading)/2);
  //calculate averages for left and right side
  leftAverage = round((upperLeftReading + lowerLeftReading)/2);
  rightAverage = round((upperRightReading + bottomRightReading)/2);
  //get servo motors' current positions
  horizontalCurrent = horizontal.read();
  verticalCurrent = vertical.read();
  //turn panel higher if upperAverage is significantly higher
  //and motor position smaller than 90
  if(upperAverage - lowerAverage > 25 && verticalCurrent < 80){
    vertical.write(verticalCurrent +1);
  }
  //turn panel 'lower' if lowerAverage is significantly higher
  //and motor position higher than 0
  else if(lowerAverage - upperAverage > 25 && verticalCurrent > 0){
    vertical.write(verticalCurrent -1);
  }
  //turn panel left if rightAverage smaller than leftAverage
  //and motor position below 176
  if(leftAverage - rightAverage > 25 && horizontalCurrent < 176){
    horizontal.write(horizontalCurrent + 1);
  }

  else if(rightAverage - leftAverage > 25 && horizontalCurrent > 0){
    horizontal.write(horizontalCurrent -1);
  }
  delay(20000);
}
```

Figure 8. Code that reads LDR values and moves the motors.

Before the algorithm decides when and where to move, first each of the four LDRs inputs and both of the servo motors' positions are read and then the averages of each side are calculated. Finally, a simple if-else if conditional expressions follow the input data preparation. Each of the conditional expressions includes a control in which direction should the motors move and detects whether the movement would direct the solar panel out of the set bounds.

The direction of movement is decided by comparing values of each side. Determining whether to move the solar cell left or right. That requires analysing the averages of the left and right side, where the average of the left side is calculated as:

$$\frac{upperLeftResistor + bottomleftResistor}{2}$$

and the same applies to the right side, but using upper right and bottom right photoresistor. The same approach is taken to regulate the vertical movement, but averages are taken from the upper and lower photoresistors. For example,

$$\frac{bottomRightResistor + bottomLeftResistor}{2}$$

is used for the bottom average. Furthermore, the bounds are respectively set 0-176 degrees and 0-80 degrees for the horizontal movement and vertical movement of the stepper motors respectively. Finally, after LDR values go below a certain boundary, the device resets its position to the start position and activates again when the sensor values go above that certain point.

3.4.2 Solar panel output algorithm

Reading of the solar panel output current is achieved with the Adafruit's current sensor INA219. The data from the current sensor is logged to the SD card using the code segment that shows in Figure 9. That includes and is limited to the sample number, electrical current in milliamperes, load voltage and time in seconds from the start of the measurements with all of them being separated with a ";".

```
void sdCardWrite(){
   myFile = SD.open("BATT.txt", FILE_WRITE);
   if (myFile) {
     myFile.print(sample);
     myFile.print(";");
     myFile.print(current_mA);
     myFile.print(";");
     myFile.print(loadvoltage);
     myFile.print(";");
     myFile.print(timeN);
     myFile.println();
     myFile.close();
 } else {
   // if the file didn't open, print an error:
   Serial.println("error opening BATT file");
 }
```

Figure 9. Method responsible for logging the data to a SD card.

Adafruit has developed a library specifically for the usage of the INA219 current sensor that is user-friendly and is ready to provide the real values out of the box without requiring any calibration.

4. Prototype

During this thesis, a prototype was developed to evaluate the proposed system. This section will introduce the main parts of the prototype.

4.1 Hardware

Hardware part of this project is made up of nine components: two servo motors, four photoresistors, an SD card shield, a solar panel and a current sensor.

4.1.1 Servo motors

The two servo motors are used to point the solar cell in the most optimal direction. Both motors are connected to the Arduino Uno using digital inputs with ports 8 and 9 (illustrated in Figure 10). The former input being used to control the cell in a horizontal direction and the latter changes the cell's vertical angle.

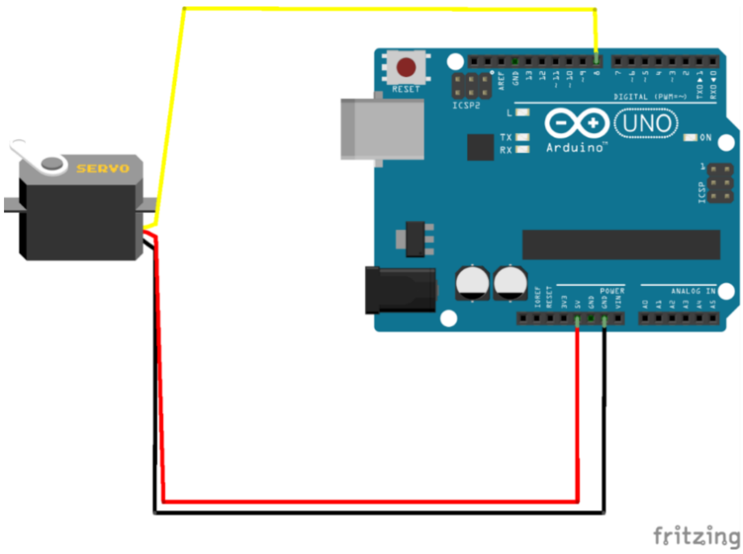

Figure 10. Servo Motor Connection to the Arduino Uno Using Port 8

Servo motors used in this thesis can move from 0 to 180 degrees. However, the motor controlling vertical movement is limited to 80 degrees since the sun does not rise higher than that angle.

4.1.2 Photoresistors

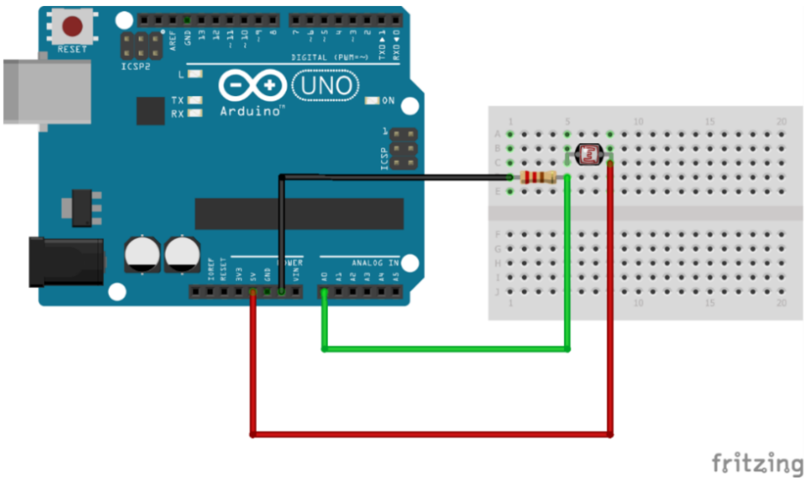

Figure 11. Photoresistor Connection to Analog Input A0

Photoresistors are fitted into each corner of the solar cell and all of them are connected to the board using analog inputs A0 (illustrated in Figure 11), A1, A2, A3 to give data regarding the light intensity in each location. Depending on the analog input values, the software then decides whether to turn the cell left or right and higher or lower.

4.1.3 Solar Panel

The Solar panel is described as a device that is able to produce electricity using the Photvoltaic effect of the energy of light. The solar panel used in this thesis is a 6V 1W mini solar panel ordered from eBay. It was mounted on a structure that is created during this thesis and the electrical current measured using a current sensor.

4.1.4 SD Card Shield

This thesis uses a shield with the Arduino Mega ADK that includes an SD card reader. The SD card reader functionality was implemented in order to log data provided by the current sensor. The SD card shield uses SPI interface to communicate with the Arduino Mega ADK.

The data that is obtained by reading the value from the solar panel and logged almost every second onto a 16GB SD card that is inserted in the module. This requires initialization of the SD card in the setup method and the use of the built-in SD library.

4.1.5 Current sensor

In this experiment we deployed Adafruit's INA219 current sensor to measure the output power from the solar panel. The INA219 current sensor is very suitable for beginners to use because they have developed a library for Arduino where they already do all the calibration and the math necessary for correct readings. Also, the INA219 boasts an accuracy of 0.1 milliamperes meaning that measurements taken are very accurate and trustworthy [16].

4.2 Experimental results

Experimental results were obtained only on two consecutive days due to time, location and weather constraints. The 3rd and 4th of May were chosen as the dates for gathering data for the fixed and moving panel setup. Python library Matplotlib was used to create graphs.

4.2.1 Weather conditions

Weather conditions for both testing days were nearly perfect, but as Tables 2 and 3 demonstrate, the second day of data gathering had marginally better weather when it comes to cloud coverage. Moreover, during peak sunlight hours the weather conditions were more or less the same and are comparable.

Table 2. Weather conditions from 8AM to 9PM on the 3rd of May, 2017 in Tartu, Estonia [17].

Time (EEST)	Temperature	Visibility	Conditions
8:00 AM	6 °C	20 km	Clear
9:00 AM	8 °C	20 km	Partly Cloudy
10:00 AM	9 °C	20 km	Clear
11:00 AM	10 °C	20 km	Clear
12:00 PM	11 °C	20 km	Partly Cloudy
1:00 PM	12 °C	20 km	Clear
3:00 PM	13°C	20 km	Clear

4:00 PM	13 °C	20 km	Clear
5:00 PM	13 °C	20 km	Clear
6:00 PM	12 °C	20 km	Partly Cloudy
9:00 PM	8 °C	20 km	Partly Cloudy

Table 3. Weather conditions from 8AM to 9PM on the 4th of May, 2017 in Tartu, Estonia [18].

Time (EEST)	Temperature	Visibility	Conditions
8:00 AM	6 °C	20 km	Clear
9:00 AM	8 °C	20 km	Partly Cloudy
12:00 PM	11 °C	20 km	Partly Cloudy
1:00 PM	11 °C	20 km	Clear
3:00 PM	12 °C	20 km	Partly Cloudy
4:00 PM	12 °C	20 km	Clear
5:00 PM	12 °C	20 km	Clear
6:00 PM	12 °C	20 km	Partly Cloudy
7:00 PM	12 °C	20 km	Clear
9:00 PM	10 °C	20 km	Partly Cloudy

4.2.2 Fixed solar panel results

Fixed solar panel measurements were taken on the 3rd of May, 2017 in order to set a baseline. The data harvesting started at 8:33 in the morning and measurements were taken for a total of 12 hours. The X-axis and Y-axis in Figure 12 respectively show time in minutes from the start of the data gathering and the output current from the solar panel in milliamperes. The blue line represents the raw data and the red line uses the moving average to smooth the graph. Mainly along the blue line, there are sudden decreases in the output that are caused by clouds covering the sun dramatically lowering the output of the solar panel.

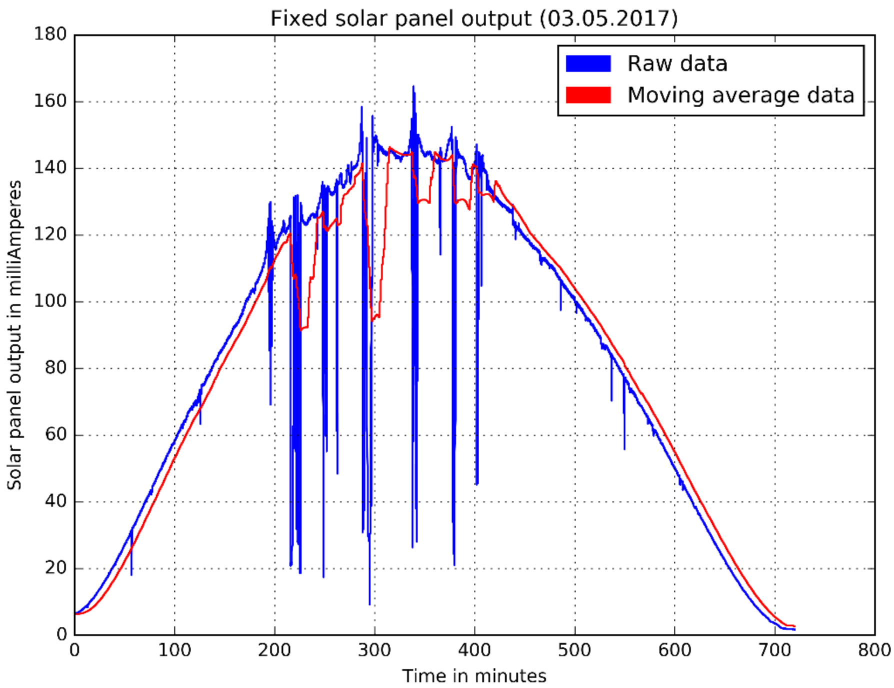

Figure 12. Fixed solar panel output graph.

The average output of the fixed panel is 82.74 milliamperes which was calculated using the moving average in order to eliminate outliers. The calculated average of the raw data was 82.69 milliamperes. The maximum output of the solar panel was 164.6 milliamperes and a minimum of 1.6 milliamperes.

4.2.3 Moving panel results

As noted above, the data gathered for the graph depicted in Figure 13, was taken on the 4th of May. Data gathering started 8:33 in the morning and lasted 12 hours as well in order to align the data with the fixed panel setup. The X-axis shows minutes from the start time and the Y-axis shows the electrical current in milliamperes produced by the solar panel for each time point. The blue line is the raw data that was logged using the Arduino Mega ADK and the red line illustrates the raw data that has been smoothened using the moving average method. However, due to calibration necessities and final fixes, the first 200 minutes are erratic that were caused primarily by resetting the position of the machine in order to achieve the best parameters.

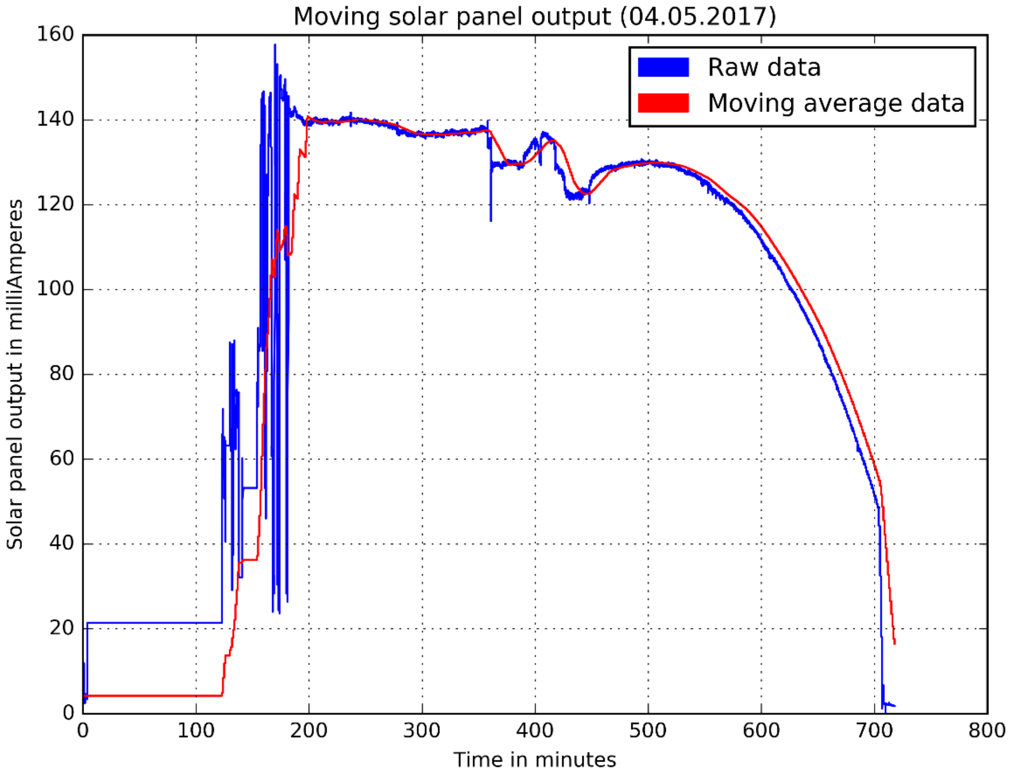

Figure 13. Moving solar panel output graph.

When the raw data is examined, the moving panel achieved an average of 99.35 milliamperes of electrical current produced with the maximum and minimum output being 157.72 and 0.0 milliamperes respectively. After smoothing with the moving average method, an average of 96.18, a maximum of 140.89 and a minimum of 6.19 milliamperes of electrical current measured.

4.3 Data analysis

Table 4 below shows the comparison between the output of the solar panel whether it was in a fixed position or moved dynamically. All the figures displayed in the table are in milliamperes.

Table 4. Comparison of fixed and moving panel output

	Raw average/Moving average (mA)	Raw maximum/Moving maximum (mA)	Raw minimum/Moving minimum (mA)	Moving average in the last 4 hours (mA)
Fixed panel	82.69/82.74	164.6/146.4	1.6/2.62	56.73
Moving panel	99.35/96.18	157.72/140.89	0.0/6.19	104.89

As noted in the previous section, the data for the moving solar panel is a bit malformed, but nonetheless, the moving averaged data shows that there is a 16% increase moving panel. Furthermore, with raw data the average increases a further 4%, thus altogether an increase of 20%.

Moreover, during earlier and later hours of the day, the moving panel generated more energy than the fixed panel. For example, if looked at the last 4 hours of data points and taken the average, the difference is phenomenal – fixed panel produced an average of 56.73 milliamperes whereas the moving panel maintained an average of 104.89 milliamperes of current produced. The contrast deepens, even more, when data from the last 3 hours of the day is taken – during the last quarter of the 12 hours, the average current produced by the moving panel is 2.3 times higher than what the fixed panel managed to produce in the same time frame. This means that in the real world when such a system is applied to a private home with no access to the commercial electrical grid network, the occupants can start using appliances and tools earlier in the morning. Furthermore, at the end of the day, the batteries are more fully charged and the chance of batteries becoming empty is greatly reduced. Moreover, the setup offering the residents a peace of mind and reducing the need for supplementing power generation with generators or similar equipment.

When it comes to the extremums of the data set, the fixed panel peaked higher than the moving panel, 164.6 and 157.72 milliamperes respectively, exposing a flaw in the prototype. Upon further inspection, the prototype correctly followed the Sun along the horizontal axis, but the mechanisms that were responsible for vertical movement did not operate as intended resulting in a lower peak current. This was likely caused by the algorithmic component where the upper and the lower side average difference was set too high and to the algorithm, it seemed that the panel was in the most optimal position.

5. Conclusion and future work

The aim of this thesis was to design and build an autonomous solar tracking system based on the Arduino microcontroller.

A prototype was designed and built on the Arduino microcontroller that tracked the Sun and greatly increased the output of the solar panel. The device can track the sun on the horizontal axis, however, vertical tracking remains an issue. Across the whole 12-hour testing periods, the tracking solar panel produced 20% more electrical current on average. Furthermore, in latter hours, the tracking panel produced on average 2.31 times more than the fixed panel.

There are several potential future perspectives in relation to this thesis. Firstly, the most pressing improvement being vertical movement detection, which would allow extracting more power out of a solar panel. Secondly, further development and testing should include additional testing periods in diverse kinds of weather conditions to more accurately determine the benefits of a solar tracking system. Thirdly, the current tracking system that was developed requires power from an outside source, meaning that in the future, it should be converted to run on a battery and the device should be powered by itself. Lastly, a container or a case should be designed and built to ensure sheltering from the elements as the current system is susceptible to rain or any kind of moisture.

References

[1] R. Leiner, “Solar radiation and water for emission-free marine mobility,” in *Energy Conference (ENERGYCON), 2014 IEEE International*, 2014, pp. 1425–1428.

[2] Wikipedia, “Solar power --- Wikipedia{,} The Free Encyclopedia.” 2017.

[3] H. Mousazadeh, A. Keyhani, A. Javadi, H. Mobli, K. Abrinia, and A. Sharifi, “A review of principle and sun-tracking methods for maximizing solar systems output,” *Renew. Sustain. energy Rev.*, vol. 13, no. 8, pp. 1800–1818, 2009.

[4] I. E. Agency, “Technology Roadmap Solar Photovoltaic Energy,” *IEA, [Online]. Available http//www.iea.org/publications/freepublications/publication/TechnologyRoadmapSolarPhotovoltaicEnergy_2014edition.pdf*, 2014.

[5] J. Rizk and Y. Chaiko, “Solar tracking system: more efficient use of solar panels,” *World Acad. Sci. Eng. Technol.*, vol. 41, pp. 313–315, 2008.

[6] W. Liu, “Sun Tracker: Design, Build and Test,” in *Vehicular Technology Conference (VTC Spring), 2015 IEEE 81st*, 2015, pp. 1–5.

[7] P. Sharma and N. Malhotra, “Solar tracking system using microcontroller,” in *Non Conventional Energy (ICONCE), 2014 1st International Conference on*, 2014, pp. 77–79.

[8] P. D. Kale, N. R. Patel, G. N. Raut, and R. B. Lanjewar, “Design and implementation of PSoC microcontroller based photovoltaic system,” in *Circuit, Power and Computing Technologies (ICCPCT), 2014 International Conference on*, 2014, pp. 1257–1262.

[9] L. Arduino, “Arduino - FAQ,” *Arduino,[Online]. Available: https://www.arduino.cc/en/Main/FAQ*, 2015.

[10] L. Arduino, “Arduino Products Compare,” *Arduino,[Online]. Available: https://www.arduino.cc/en/Products/Compare*, 2015.

[11] L. Arduino, “Arduino introduction,” *Arduino,[Online]. Available: http://arduino.cc/en/guide/introduction*, 2015.

[12] L. Arduino, “Arduino - Builtin Examples,” *Arduino,[Online]. Available:*

https://www.arduino.cc/en/Tutorial/BuiltInExamples, 2015.

[13] L. Arduino, “Arduino - SPI,” *Arduino,[Online]. Available: https://www.arduino.cc/en/reference/SPI,* 2015.

[14] “Global Horizontal Irradiation Estonia,” *Solargis-Estonia-GHI-solar-resource-map-en.png*. Solargis.

[15] W. Commons, “Simplified map of sunshine duration in Europe according to national data.” 2010.

[16] A. Industries, “INA219 High Side DC Current Sensor Breakout - 26V �3.2A Max,” *adafruit industries blog RSS*. .

[17] “Weather History for 26242 - May, 2017,” *Weather Underground (10.226.235.230)*. https://www.wunderground.com/history/wmo/26242/2017/5/3/DailyHistory.html?req_city=Tartu&req_state=78&req_statename=Estonia&reqdb.zip=00000&reqdb.magic=1&reqdb.wmo=26242&MR=1.

[18] “Weather History for 26242 - May, 2017,” *Weather Underground (10.226.233.254)*. https://www.wunderground.com/history/wmo/26242/2017/5/4/DailyHistory.html?req_city=Tartu&req_state=78&req_statename=Estonia&reqdb.zip=00000&reqdb.magic=1&reqdb.wmo=26242.

Appendix

I. License

Non-exclusive licence to reproduce thesis and make thesis public

I, Karl Lubja,

1. herewith grant the University of Tartu a free permit (non-exclusive licence) to:
   1.1. reproduce, for the purpose of preservation and making available to the public, including for addition to the DSpace digital archives until expiry of the term of validity of the copyright, and
   1.2. make available to the public via the web environment of the University of Tartu, including via the DSpace digital archives until expiry of the term of validity of the copyright,

   Energy Harvesting Optimization with Solar Panels,

   supervised by Mohan Liyanage and Satish Srirama,

2. I am aware of the fact that the author retains these rights.
3. I certify that granting the non-exclusive licence does not infringe the intellectual property rights or rights arising from the Personal Data Protection Act.

Tartu, **11.05.2017**