Technical University Munich

Project report

Humanoid sensors and actuators

Light-following solar panel

Author:
David Lichtenwalter

Supervisor:
Florian Bergner

July 16, 2018

Technische Universität München

Contents

1 Introduction **2**
1.1 Motivation 2
1.2 Project approach 2

2 MCU and development toolchain **2**

3 Mechanical setup **2**
3.1 2-axis pan and tilt arm 2
3.2 Panel mount 3

4 Electrical circuit **4**
4.1 LDR array 4
4.2 Solar panel 5
4.3 LCD 6
4.4 Servo motors 6
4.5 Tolerance potentiometer 6
4.6 The finished solar tracker module 7

5 Tracking algorithm **7**

6 Conclusion and discussion **8**
6.1 Results 8
6.2 Possible improvements 9

1 Introduction

1.1 Motivation

The power output of solar panels heavily relies on the angle of the sun rays impacting the panel. To maximize power output, we want to keep this angle as close as possible to 90° at all times. Studies have shown the importance of correct orientation and tilt of solar panels: minor deviations from the optimal orientation of the panel can already result in a power loss of up to 49% [5].

1.2 Project approach

In the following, I present a small two-axis solar panel mount that automatically orients the solar panel vertically and horizontally towards the direction of the brightest light source in the environment. Sensing the orientation of the light source is done by an array of 4 light-dependent resistors that are placed in a separator structure. The sensitivity of the actuation can be controlled via a potentiometer. The resulting generated voltage of the panel is being measured and displayed together with the sensitivity parameter on a LCD.

2 MCU and development toolchain

In order to drive the system, an ATMega32 micro-controller mounted on a breadboard was used with a clock frequency of $8MHz$. The entire project was developed on MacOS 10.13 and Xcode 9 with the help of the *CrossPack AVR* toolchain [4] and the Xcode plugin *xavr* [3].

The following command for setting the fuse bits `0xC4` and `0xD9` was executed to configure the ATMega32 for this project:

```
avrdude -U lfuse:w:0xc4:m -U hfuse:w:0xd9:m
```

3 Mechanical setup

3.1 2-axis pan and tilt arm

Since we want to move the panel mount both vertically and horizontally, we need one actuator for each degree of freedom, i.e. two motors. Servo motors are best suited for this purpose, since they can be easily and precisely set to a certain position via PWM without any additional drivers needed. The range of 180° is not a limitation: the sun rises in the east and sets in the west. The two servo motors were mounted on a pan-tilt-kit with an end-effector that is usually used for mount camera equipment. The pan-tilt-kit was bought from Amazon [1].

3.2 Panel mount

In order to mount the solar panel to the servo-controlled arm and provide the mechanical structure for the photosensors, I designed an appropriate CAD model and printed it with a 3D printer. The rendered CAD parts can be seen in figures 1 and 2.

The panel platform has four holes in the middle that are used to screw it to the end-effector adapter. Another 5 holes are used on top of the platform to allow for the cables of the LDRs and the solar panel to go through.

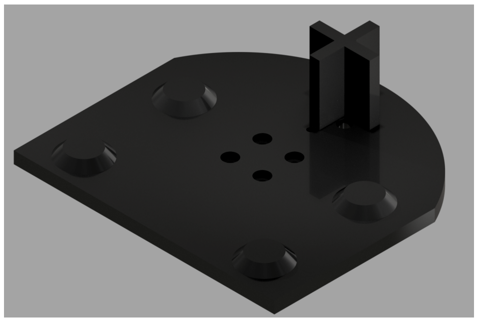
Figure 1: Rendering of the solar panel and sensor mount

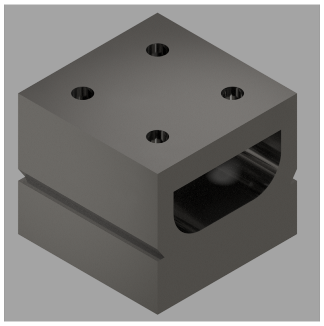
Figure 2: Rendering of the adapter for the end-effector

Separator

Additionally, we have the separator, which consists of a cross that is raised $3cm$ from the platform. The separator makes the estimation of the light source location possible while still having the LDRs close together.

Another option would have been to place an LDR on each corner of the platform. However, I decided to go with the solution above, since it heavily simplifies the cable management of the system. Also, I believe the separator method to be more accurate, since the differences in resistance between the LDRs are more severe, therefore diminishing the influence of resistance tolerances.

The fully assembled solar panel mount module with all the connectors is shown in the two photos below.

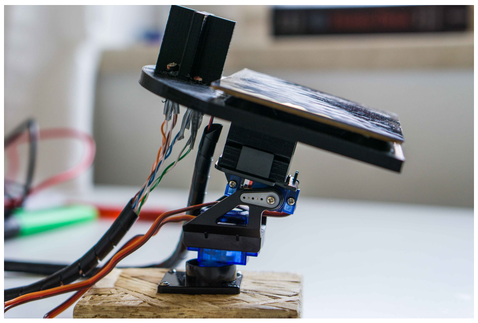
Figure 3: Side view of the panel mount module

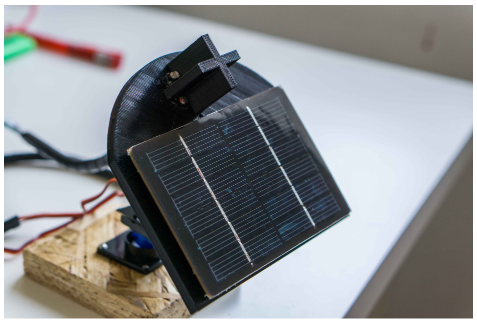
Figure 4: Top view of the panel mount module

4 Electrical circuit

In figure 5, we see the entire schematic of setup with all the peripheral systems connected to the ATmega32.

4.1 LDR array

To estimate the location of the brightest light source relative to the system, an array of four light dependent resistors (LDR) was used. LDRs change their resistance value depending on the amount of light they are exposed to. The brighter the the light source, the smaller the resistance value. The resistance value of the LDRs used in this project have approximately the following range of operation:

- $R_{low} \approx 100\Omega$ in a very bright environment
- $R_{high} \approx 200k\Omega$ in a very dark environment

Each LDR is arranged in the corner of a $3cm$ high separator on top of the panel mount. This separator acts as light shielding on order to better differentiate between the ADC values. The LDRs are connected to ground over a resistor $R_{1,2,3,4} = 1.3k\Omega$ and the voltage is measured with the ADC channels $0-3$, therefore acting as a voltage divider. The extreme values of the LDRs then translate to the following ADC voltages:

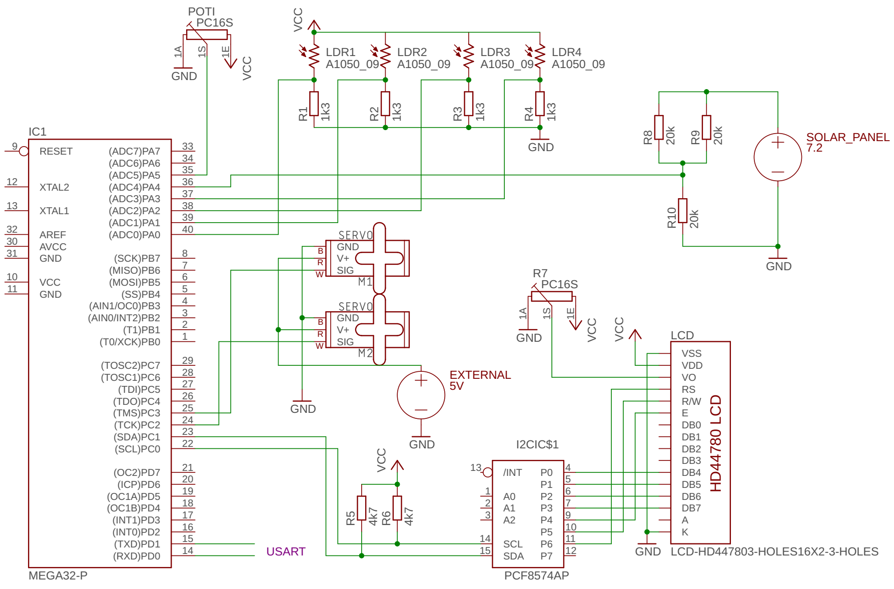

Figure 5: Complete schematic and I/O connections

$$V_{\text{ADC, max bright}} = \frac{5V \cdot R_{low}}{R_{low} + 1.3k\Omega} = 4.643V \tag{1}$$

$$V_{\text{ADC, max dark}} = \frac{5V \cdot R_{high}}{R_{high} + 1.3k\Omega} = 0.032V \tag{2}$$

Therefore, we exploit almost the full $0 - 5V$ range of the ADC.

4.2 Solar panel

The current voltage generated by the solar panel is measured with the ADC and displayed on the LCD. Since the solar panel has a peak voltage of $7.2V$, the voltage has to be scaled down to the $0 - 5V$ range with an appropriate voltage divider (see figure 5, $R_8 = 20k\Omega$, $R_9 = 20k\Omega$, $R_{10} = 20k\Omega$) to prevent peak voltages from damaging the MCU port. With the values above, the peak voltage at the MCU port is given as

$$V_{peak,ADC4} = 7.2V \cdot \frac{20k\Omega}{10k\Omega + 20k\Omega} = 4.8V \tag{3}$$

4.3 LCD

In order to give feedback to the user without any additional PC necessary, a 16×2 type HD44780 LCD is used to display the measured solar panel voltage and the tolerance/sensitivity of the sensor array that is controlled with the potentiometer. The LCD is driven with a PCF8574A I^2C backpack, which makes cable management way easier, since we only need two data connections for the I^2C, SDA and SCL. Pull-up resistors of the SDA and SCL pins are enabled.

Since driving a LCD is not really part of sensing or actuation, I decided to use an existing library to interface the LCD with I^2C to spend more time on tuning the tracking algorithm and the parameters. The library that was used is by *Davide Gironi* [2].

One problem I encountered is that writing onto the LCD takes a lot of time. This is party due to the I^2C communication and maybe even due to inefficient implementation of the library. Displaying the current values on the LCD with a high rate (e.g. $50Hz$) messed up the light source tracking. Therefore, I chose to just update the LCD two seconds, i.e. with a frequency of $0.5Hz$.

4.4 Servo motors

The servo motors are driven with over a GPIO port and the PWM signal is generated by manually switching the pin output of `PORTC` with the 16-Bit `TIMER1`. `servo1` is the servo on the bottom that controls the horizontal movement, while `servo2` controls the vertical movement.

As it can be seen in the `servolib.h` header file, with the implemented algorithm we can actually drive 8 servo motors at the same time by sequentially switching the output pins of `PORTC` in the interrupt service routine. In our case, only two servo motors (on `PC2` and `PC3`) are driven.

4.5 Tolerance potentiometer

With the potentiometer, we can set the sensitivity of the system according to the current ambient situation. Sensitivity has been implemented via a tolerance value in the tracking algorithm. This is being explained further in section 5. Here, we map the 10-bit ADC input of $0 - 1023$ to a tolerance parameter that ranges from 30 to 250.

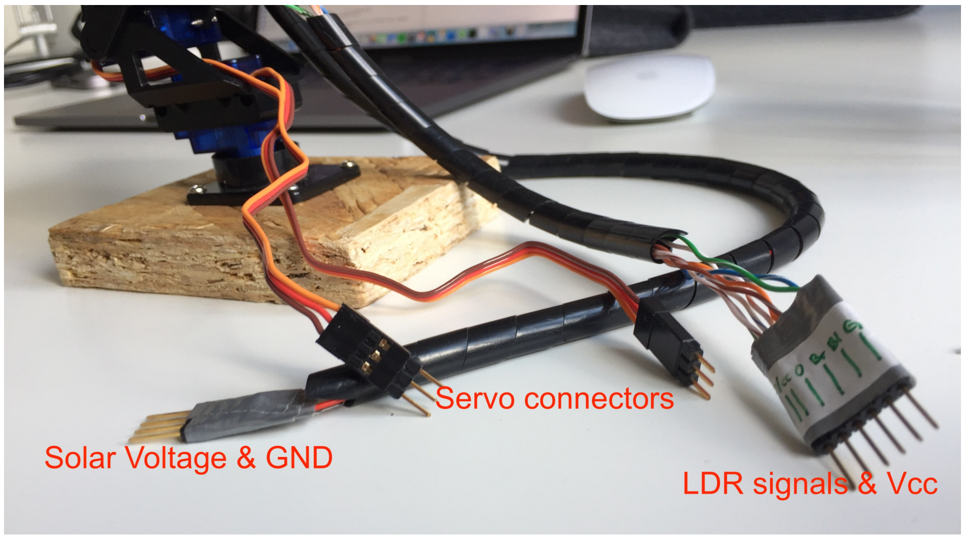

Figure 6: The connectors of the solar tracking module

4.6 The finished solar tracker module

In the end, we have an independent solar tracker module that can be easily detached and connected to any MCU circuit. It provides simple connectors for each pin we need for the sensing and actuation.

We have one connector that contains the signal for each LDR as well as a VCC pin which should be connected to the MCU's supply voltage. The second custom connector is for measuring the solar voltage, i.e. containing a GND pin and a V_{Solar} pin. Additionally, we obviously have the two servo connectors. The connectors can be seen in figure 6.

5 Tracking algorithm

The idea of the tracking algorithm is to take into account the differences between each LDR value and move the servos according to their relative position in the LDR array.

In the following pseudocode listing, the the very basic idea of the tracking algorithm is showcased.

```
// Main loop
while True:

  values = ReadLDR()     // Get the adc data
```

```
averages = CalculateAverages(values)

// Move the vertical servo depending on the difference top to
  bottom
if average_top > average_bottom:
  MoveVerticalServoUp()
else
  MoveVerticalServoDown()

// Move the horizontal servo depending on the difference right
   to left
if average_left > average_right:
  MoveHorizontalServoLeft()
else
  MoveHorizontalServoRight()
```

We see, that once the average value on the top, i.e. the average of the two LDRs on the top, is higher than the average value on the bottom, we assume the light source is coming from the top direction. Therefore, we move the vertical servo motor up until the averages are roughly equal again.

Obviously, it is also important to implement a tolerance value that prevent the servos from changing their position when there is only a small fluctuation in the values. This tolerance value is being controlled by the before-mentioned potentiometer to give the user to ability to adjust the system to the current environment.

6 Conclusion and discussion

6.1 Results

We see that the solar panel in fact follows a changing position of a light source quite quickly. Placed right next to a window, it will orient itself towards the direction of the window immediately. Also, in a sufficiently neutral environment it will follow a moving flashlight with ease and orient the panel in a 90 degree angle towards the light source.

One should note that the sensitivity of the system should always be tuned to the current environment: With a lot of different light sources present, a lower tolerance parameter (more sensitive) is recommended since the difference in the signals might not be enough otherwise. However, a less sensitive setting makes the panel always more stable.

6.2 Possible improvements

After attaching the 3D-printed panel mount and the solar panel itself, the weight turned out to be quite heavy for the two little servos. While they still work rather well, the dynamics with a cardboard mockup panel were definitely better. Now, with the heavier load, we cannot allow quick movements of the servos since the arm then starts to swing back due to the unstable pan-tilt-kit. However, considering the fact that in the real world, the sun does not quickly change its position, this mechanical problem does not matter too much.

References

[1] (2018). Pan-Tilt Kit on Amazon. `https://www.amazon.de/gp/product/B077S3YZWD/ref=oh_aui_detailpage_o00_s00?ie=UTF8&psc=1`. [Online; accessed 4-July-2018].

[2] Davide Gironi (2013). An AVR Atmega library for HD44780 based lcd connected through i2c. `http://davidegironi.blogspot.com/2013/06/an-avr-atmega-library-for-hd44780-based.html`. [Online; accessed 4-July-2018].

[3] jawher (2016). Github repository of the xavr Xcode plugin. `https://github.com/jawher/xavr`. [Online; accessed 4-July-2018].

[4] Objective Development (2013). CrossPack for AVR Development. `https://www.obdev.at/products/crosspack/`. [Online; accessed 4-July-2018].

[5] Pecan Street Research Institute (2013). Study on orientation of solar panels. `http://www.pecanstreet.org/2013/11/your-solar-panels-arent-facing-the-wrong-way/`. [Online; accessed 4-July-2018].