# Electronics in mechatronics

# Automated chicken coop door

Durczok Damian

Fijak Mateusz

## 1. Motivation

Automation is playing a constantly increasing important role in more and more new areas. Since the original Industrial Revolution process automation have taken not only manufacturing and production but also transport, management and services.  Recently more and more attention is paid to automating everyday chores. One of the approaches – Smart Home – assumes convenient and automated way to control and monitor home attributes such as lightning, climate, entertainment systems etc. Though this approach is developing with great rapidity it still lacks some core functionalities for households in less urbanized regions – countryside.

Our goal is to extend the Smart Home approach outside the walls of the house so that larger group of people can benefit from it.

## 2. Elements selection

**H – Bridge:**

Choosing the correct transistors is critical as these will be driving the largest load. Since the load will be a motor that may pull several amps of current, MOSFETs must be used.

To get a better idea of the parameters that will be used in this section, we will define a few parameters. The diagram may be used as a reference.

- The Gate-to-Source Voltage ($V_{GS}$) is the voltage difference between the gate and the source.
- The Drain-Source Voltage ($V_{DS}$) is the voltage difference between the drain and the source.
- The Drain-to-Source Resistance ($R_{DS}$) is the resistance between the drain and the source.

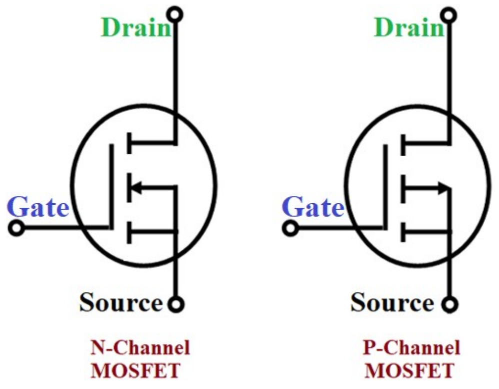

*Figure 1 – N- and P-channel MOSFETs*

How to choose a MOSFET:

1) Choose the maximum output voltage.

The Drain-Source Voltage ($V_{DS}$) should be at a value comfortably above the maximum voltage you want to switch.

2) Choose an “OFF mode” voltage.

We must find at what voltage the MOSFET will be cut-off, we will define this as “OFF mode”. The gate threshold voltage ($V_{GS\ (TH)}$) is the voltage required between the Gate and Source to turn ON the MOSFET. Anything below this value results in an “OFF mode”.

“OFF mode” $I_D = 0$ for $V_{GS} < V_{GS(TH)}$

3) Choose the maximum output current.

The Continuous Drain Current ($I_D$) should be at a value comfortably above the maximum current you want to switch. However, as the MOSFET has an ohmic region where it is not fully open at a given voltage, we need a MOSFET whose Static Drain-to-Source On-Resistance ($R_{DS(on)}$) at our maximum driving potential ( $V_{GS(max)}$ ) is low enough for the current you want to switch.

At this point it is also important to calculate the power dissipated by the MOSFET, given by $I^2 * R_{DS(on)}$. Ideally keeping this value low, for example 5W. This value may be read from the MOSFET characteristics found on a specific components datasheet.

Since our circuit will be controlled by a microprocessor, we need a logic level MOSFET with the following specifications:

$$V_{DS} > 12V,\ V_{GS(th)} < 2.5V,\ I_D > 5A\ @\ V_{GS(max)}\ \text{and}\ P < 5W$$

The following is a comparison of key parameters of a few SMD MOSFETS:

| P-Channel MOSFET | $V_{DS}$ | $V_{GS(th)}$ Max | $I_D$ | P |
|---|---|---|---|---|
| AO3401A | 30V | 1.3V | 4A @ 10V | 735mW |
| IRLML2244TRPbF | 20V | 1.1V | 4.3A @ 4.5V | 1W |
| APM4953<br>*Dual | 30V | 2V | 3.6A @ 4.5V<br>4.9A @ 10V | 1.25W<br>1.45W |
| AOD409 | 60V | 2.4V | 20A @ 5V<br>5A @ 5V | 22W<br>1.4W |
| AO4407 | 30V | 2.8V | 7A @ 5V<br>12A @ 10V | 1.5W<br>2W |

*Table 1 – SMD P-channel MOSFETs comparison*

| N-Channel MOSFET | $V_{DS}$ | $V_{GS(th)}$ Max | $I_D$ | P |
|---|---|---|---|---|
| IRLML0040TRPbF | 40V | 2.5V | 2.9A @ 4.5V<br>3.6A @ 10V | 655mW<br>725mW |
| 2N7002 | 60V | 2.5V | 0.05A @ 5V<br>0.5A @ 10V | ~0W<br>1.25W |
| AO3400A | 30V | 1.4V | 5A @ 4.5V<br>5.8A @ 10V | 900mW<br>1W |

*Table 2 – SMD N-channel MOSFETs comparison*

The two parts highlighted in green are the closest matching in parameters that fulfill the minimum requirements. Given that this is an H-bridge and only two MOSFETs should be active at a time, assuming 5A of constant power, we may expect a total power output of around 1.75W.

***Due to the availability of various parts, we replaced the AO4407 with the AO3401A for the sake of completing this project in time.*

With the most important parts of the H-Bridge selected, we may finish designing and simulate the circuit in LTSpice to check for proper operation.

The diagram on fig. 2 was designed to operate using only two inputs, enable and direction. The enable input allows current to flow through the circuit and the U1 MOSFET is responsible for this. The other direction input goes through a circuit with three NAND gates. Two of these NAND gates function as an AND while the third is to negate the signal. The reason for this is that I can order a single component with several NAND gates. The way the circuit is set-up means that only two crossing pairs of MOSFET are on at a time. Meaning the motor will spin in either one direction or the other at all times.

There is one more crucial moment, the MOSFET does not transition between states instantaneously. Meaning that during the changing of directions, there is a brief moment where there is a short circuit between $V_{CC}$ and Ground. That is why it is essential to first disable the En pin before switching motor direction.

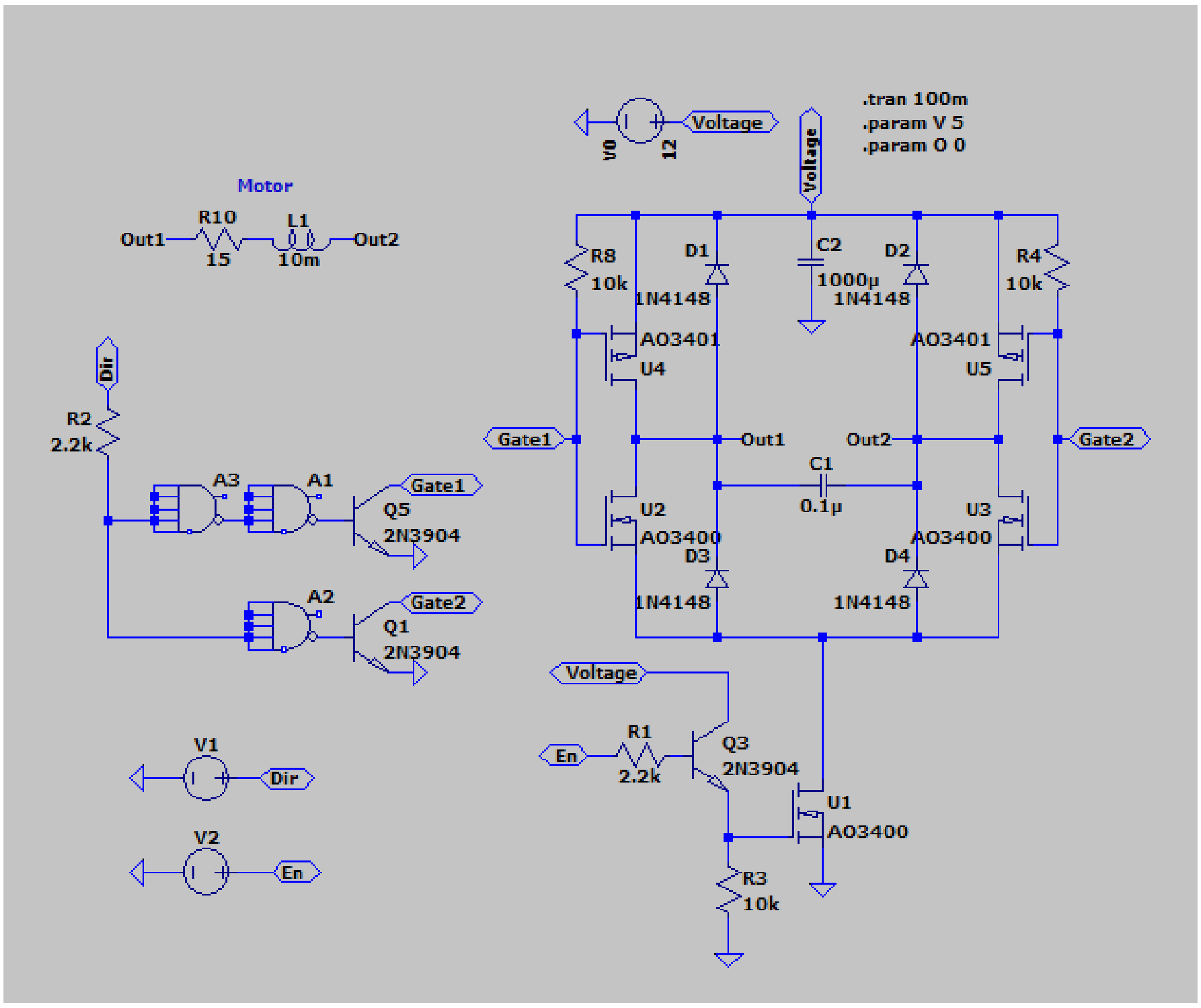

*Figure 2 – H-bridge schematic diagram*

On the plot visible on fig. 3 are the results of the simulation. The first graph plots the inputs to the system. The green plot represents the direction while the blue is the enable. We can see that during any change in the direction the enable is not powered.

The second graph shows the voltage on the gates of the MOSFETs, the right side of the H-Bridge is a negated signal of the left side. Additionally, we see noise that appears when the motor is enabled.

The final graph shows the current going through the motor, we may see that it coincides with the direction set at the input.

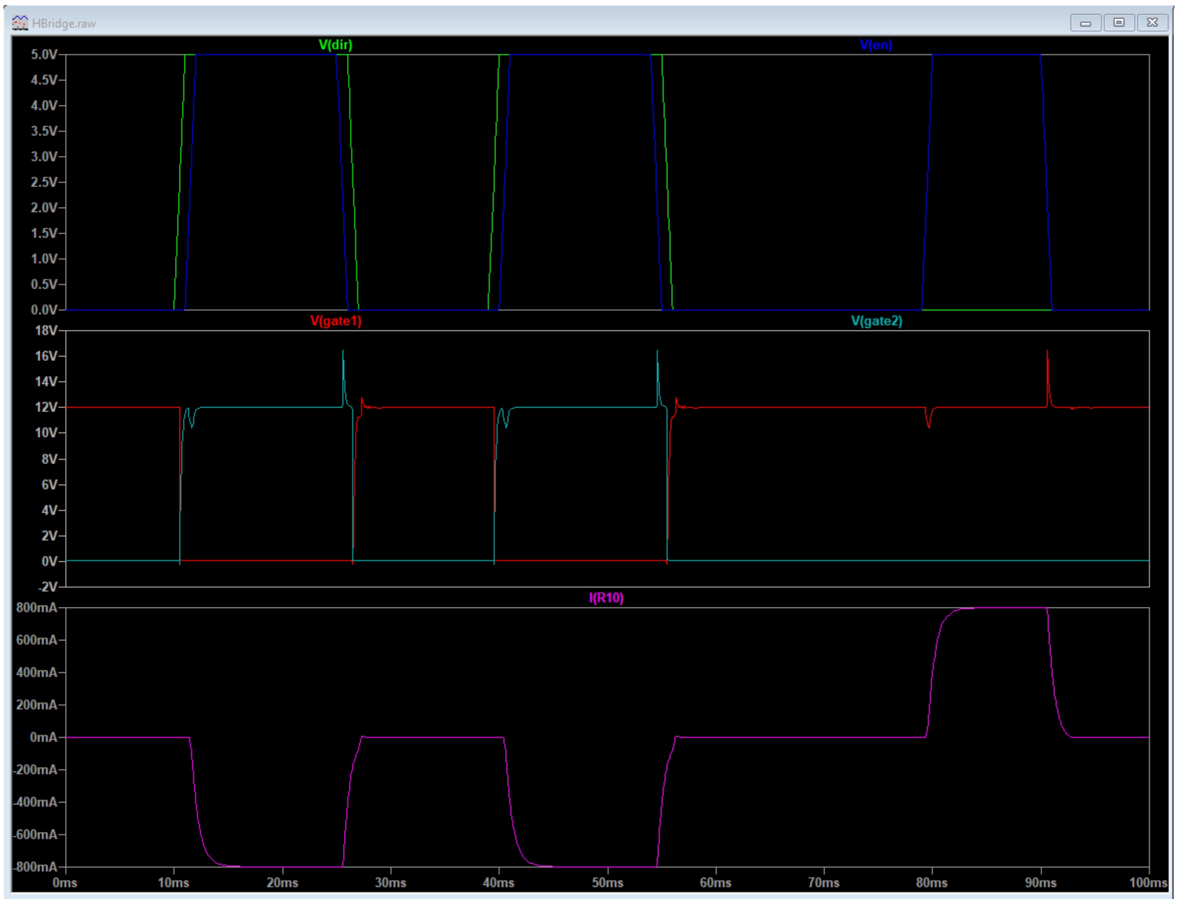

*Figure 3 – H-bridge simulateion results*

By confirming the design within LTSpice, the next step is to design the H-Bridge circuit board and carry out real tests. The program used for designing the board layout is KiCad. At first a few rapid prototypes were created. This was vital as a few mistakes where quickly identified.

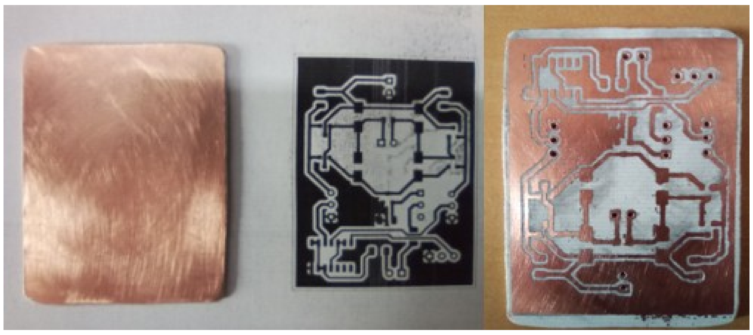

*Figure 4 – H-bridge prototype*

After a few corrections, this was the final circuit for the H-Bridge which is shown on the fig. 5.

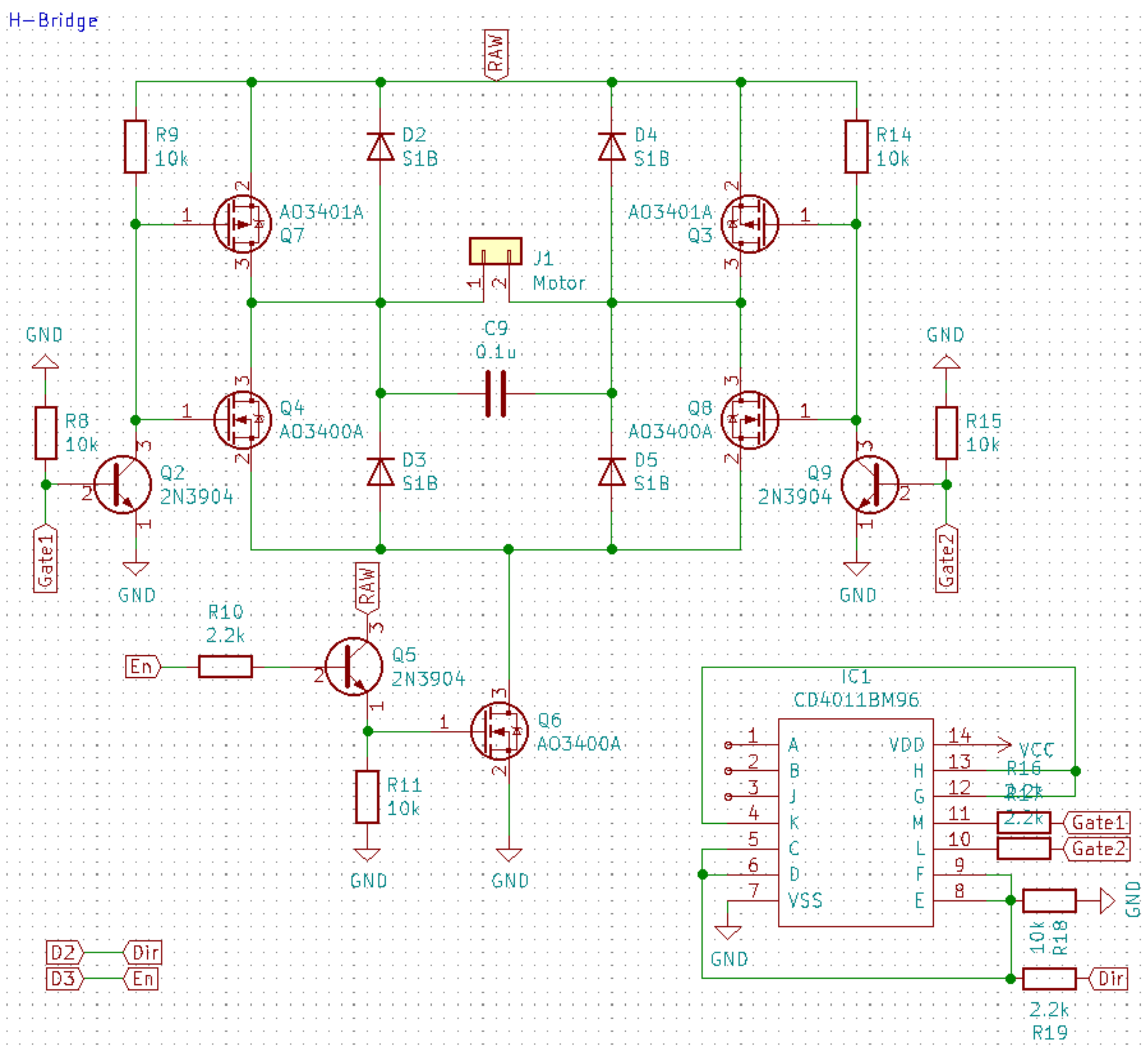

*Figure 5 – Schematic diagram of the final version of H-bridge*

**Processor:**

For the processor, we are using an ATmega328. This is a fairly common chip with a lot of online documentation and circuits. This section will cover the minimum requirements needed to get the processor working.

We are using a 16 MHz crystal and a couple of 22 pF capacitors as the external clock. The circuit is designed to also accept a resonator, in this case the two capacitors may be omitted.

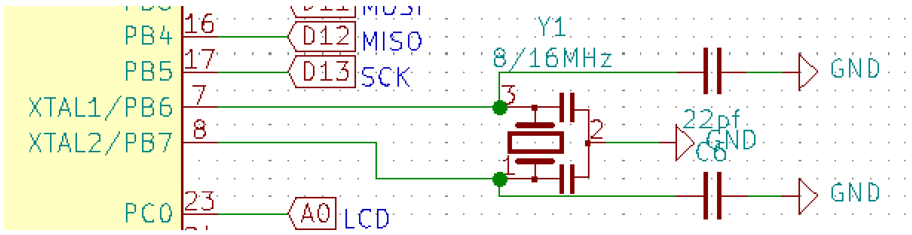

Figure 6 – Schematic diagram section highlighting the *oscillator* part

The 0.1uF capacitor and 10k pullup resistor provides a reset "pulse" when the RTS line is brought low during boot loading. An additional diode is provided to avoid voltage spikes which may potentially put the chip into "high voltage programming" mode.

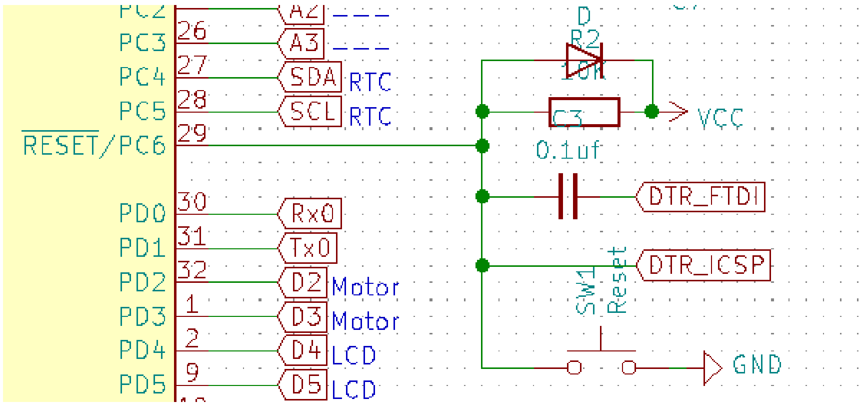

Figure 7 – Schematic diagram section highlighting the *reset* part with safety features

There are two header pinouts provided for programming. The FTDI header is used when a bootloader is loaded on the chip. If there is no bootloader, the ICSP (In Circuit Serial Programming) header is needed.

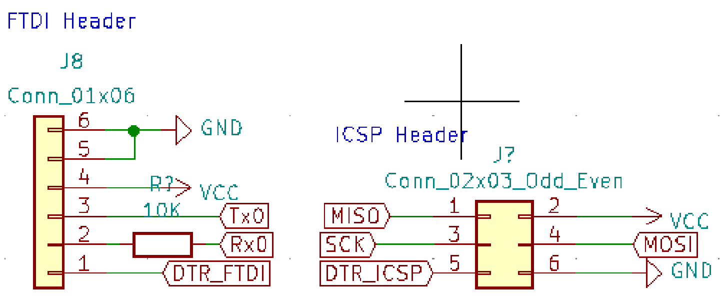

Figure 8 – Schematic diagram section highlighting part related to microprocessor programming

The processor should take around 19mA of power with the LED, this does not include all the other circuits that will be attached to this processor mentioned in the other sections.

**Power Supply:**

The power regulator is based around the mic5205, which is a very small and low-priced regulator. The capacitors are for decoupling the power line (to reduce noise). An additional LED is provided to signal whether the processor is receiving any power.

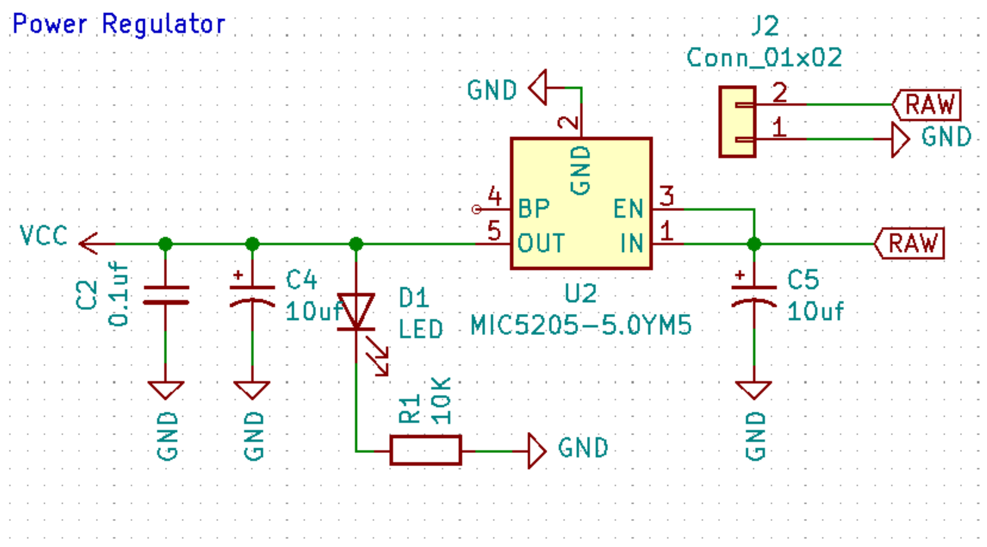

*Figure 9 – Schematic diagram of power regulator*

**Real-Time Clock:**

The RTC module is based on the DS1307 chip. The circuit itself is taken from its datasheet. Additionally, we need to attach a 3.3V battery. This is to ensure that the devices timer will work reliably even for brief power loss.

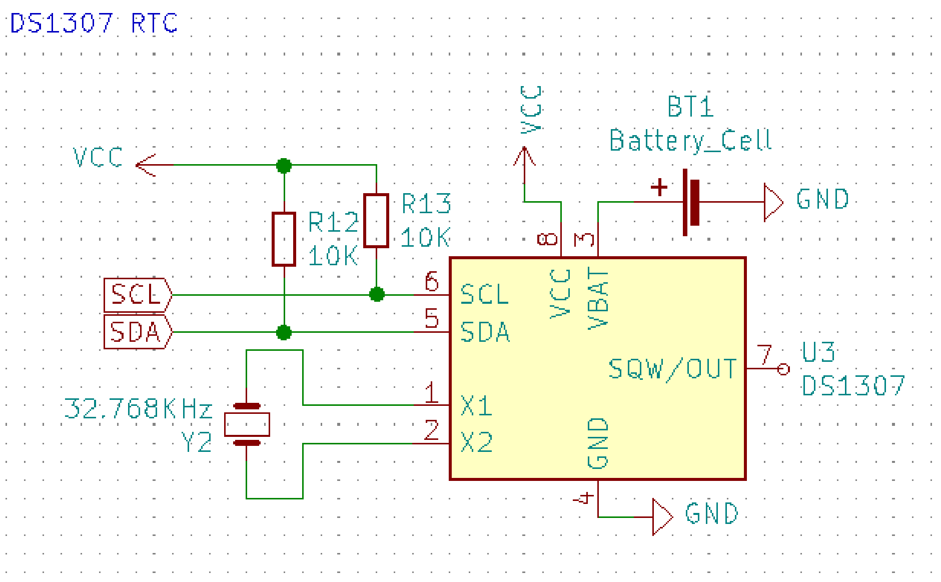

*Figure 10 – Schematic diagram of real-time clock*

**Buttons and display:**

As the display and other circuits already take a lot of the IOs of the processor, we decided to constrain the buttons to one IO. This was done using varies resistors to provide an analog signal based on the buttons pressed. We must consider that since the output depends on the precise characteristics of the resistors, temperature and even power supply instability can give false readings by providing a different voltage on the analog pin. Resistors must be chosen so that the analog signal belongs to a broader spectrum of values and compensate for any variance.

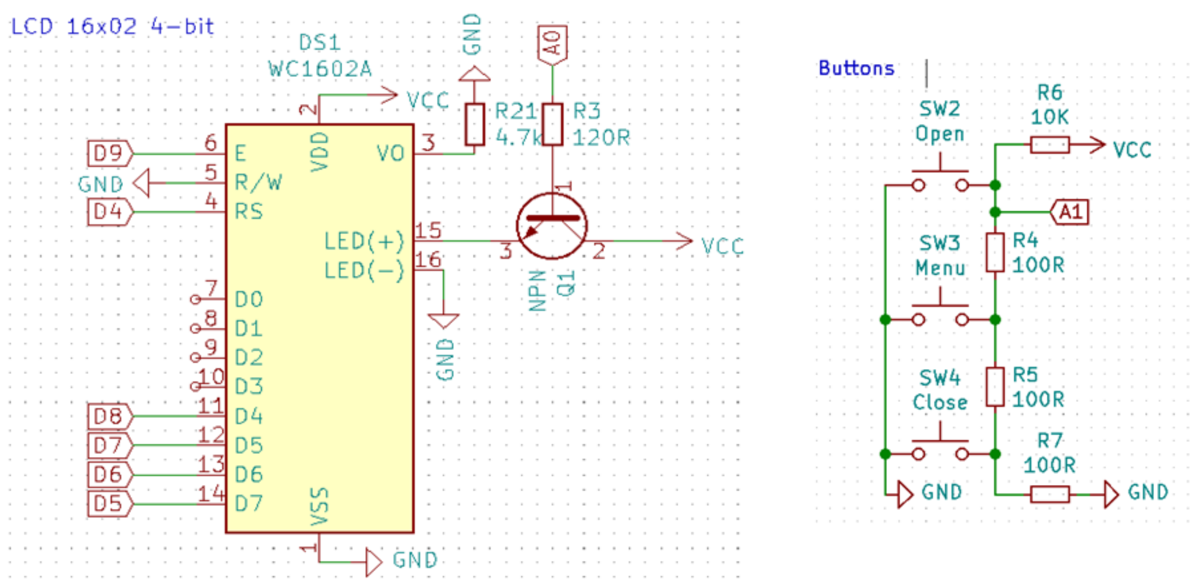

*Figure 11 – Schematic diagrams of liquid crystal display and control buttons*

## *3*. Manufacturing:

The final step is to get the board manufactured and test all the circuits as one device. Below is the front view of the layout of the PCB board.

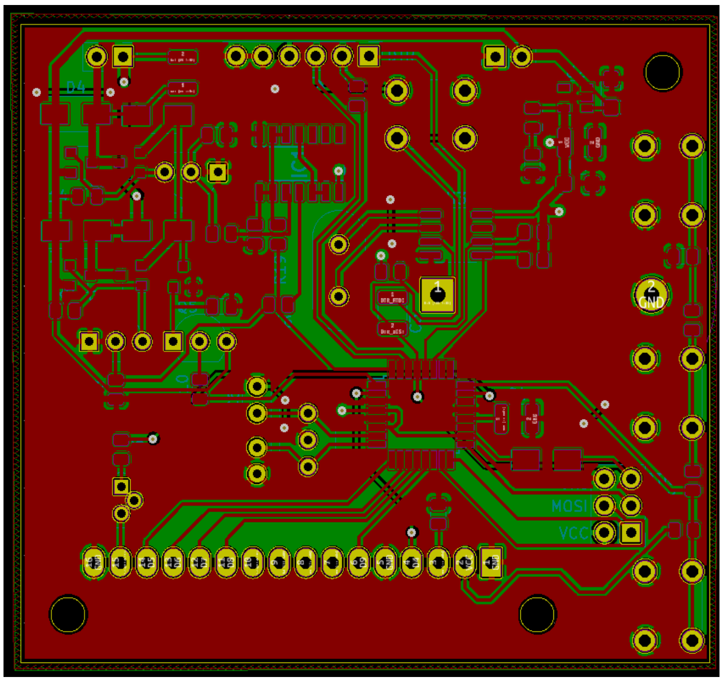

*Figure 12 – Front view of the PCB board*

The first step was to solder the minimal circuit and load the bootloader onto the microprocessor. This was done using a USBASP programmer and Arduino board.

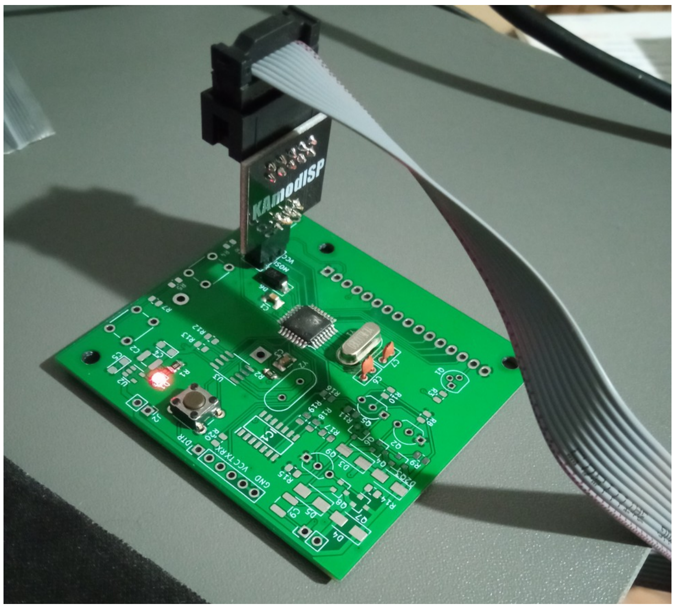

*Figure 13 – Loading bootloader onto the microprocessor*

Next the rest of the components were attached and the code loaded through the FTDI header. A few mistakes where discovered which required some modifications. This includes the wrong pinout for the transistors and mirrored MOSFETS.

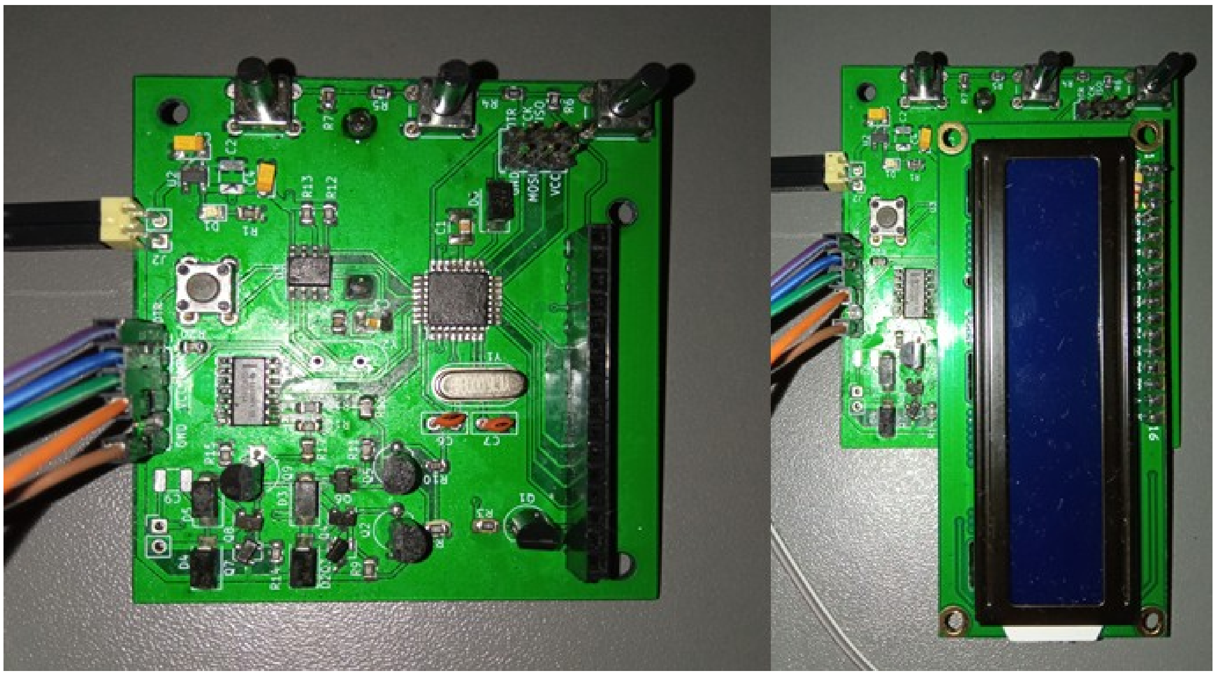

*Figure 14 – Real board testing*

The video attached demos the electronics and code provided. The picture above and video does not feature a functioning real time clock as the oscillator was mistakingly not ordered. However a version was tested with an external real time clock that proves that the code works and the circuit shouldn't have any issues.