Toni Parkkinen

Automated Solution for Tracking and Predicting Animal Growth

Thesis
Spring 2019
School of Technology
Automation Engineering
The purpose of this thesis was to design and manufacture a prototype device that could read and write information from an RFID tag, read information from the scale used by the client, and to write the data directly to an SD-card in an easy to use form. By assigning each animal its own RFID tag the client can separate the animals automatically based on their unique code number and then save the numbers directly to an SD-card together with the corresponding animal's weight information. The collected information can then be used to follow the growth of the animal by transferring the weighing data to a pre-made program that compares the changes in the animal weight during its lifetime. The finished prototype was left to the client for future development.

The thesis mostly focused on the design and manufacturing of the prototype, including also sections about the use of information gained through weighing. The idea behind this thesis was to lessen the amount of work and workers needed to collect the information.

Before the prototype can be put to use the client is going to develop a custom circuit board and a case for the machine so that it is possible to use and store the machine outdoors. The prototype is, however, functional and the wanted goals were reached within the schedule.

Keywords: automation, microprocessors, animals, machine design

Opinnäytetyön keskittyi laitteen suunnitteluun ja valmistukseen, sisältäen myös luvun lampaiden painotiedon käytöstä. Opinnäytetyön taustana oli idea lampaiden kasvun seurannan aikana tarvittavan työmäärän ja työntekijöiden vähentämisestä.

Ennen prototyypin käyttöönottoa toimeksiantajan on tarkoitus kehittää piirilevy ja kotelo laitteelle, jotta laitteen käyttö ja säilytys ulkotiloissa olisi mahdollista. Laite on kuitenkin toiminnallinen ja haluttu lopputulos saavutettiin aikataulussa.

Asiasanat: automaatio, mikroprosessori, eläimet, laitesuunnittelu
# TABLE OF CONTENTS

Thesis abstract................................................................................................................. 2
Opinnäytetyön tiivistelmä................................................................................................. 3
TABLE OF CONTENTS ..................................................................................................... 4
Tables, Figures and Pictures .............................................................................................. 6
Terms and Abbreviations................................................................................................. 7

1 Introduction .................................................................................................................... 10

2 Goal and background of the Thesis ............................................................................. 11

3 Theory of Animal Growth Tracking ............................................................................ 12
   3.1 Birth weight ............................................................................................................. 12
   3.2 Six Week weight .................................................................................................... 13
   3.3 Four Month weight ............................................................................................... 14
   3.4 Other weighings ................................................................................................. 15
   3.5 Using the results .................................................................................................. 16
   3.6 Uses of A-weight ............................................................................................... 18

4 Hardware .................................................................................................................... 21
   4.1 Selecting the parts to be used ............................................................................. 21
   4.2 Uno R3 Microprocessor ...................................................................................... 21
   4.3 ESP32 .................................................................................................................. 23
   4.4 ESUMIC LCD-Screen ......................................................................................... 23
   4.5 OLED Display ..................................................................................................... 25
   4.6 MicroSD, SD-Card Reader ................................................................................... 25
   4.7 RC522 RFID Reader ........................................................................................... 26
   4.8 EM-18 RFID module .......................................................................................... 27
   4.9 MAX3232 RS232 Adapter ................................................................................... 28

5 Electrical Planning ....................................................................................................... 29

6 Programming ................................................................................................................ 32
   6.1 Setup .................................................................................................................... 33
   6.2 Date select .......................................................................................................... 33
   6.3 Loop ..................................................................................................................... 34
6.4 Mode select .............................................................................................................. 34
6.5 Read and write data ............................................................................................ 34
6.6 Manual and automatic reading of tags ................................................................. 34
6.7 Collecting the weight data from the animals ....................................................... 35
6.8 Testing .................................................................................................................... 35

7 Conclusion ................................................................................................................ 36

BIBLIOGRAPHY ........................................................................................................ 37
Tables, Figures and Pictures

Table 1. Example weight table ................................................................. 16
Table 2. Example table of data for choosing the breeding sheep .......... 19
Table 3. Sheep index values ................................................................. 20
Table 4. LCD screen pins ................................................................. 24

Figure 1. Arduino and LCD screen module ........................................ 29
Figure 2. Arduino, RFID and MicroSD module .................................. 30
Figure 3. Arduino and RS232 module ................................................ 30
Figure 4. Arduino and 4 buttons ......................................................... 31
Figure 5. Programs logic loop ............................................................ 32

Picture 1. Arduino IDE ................................................................. 10
Picture 2. A 3-day old lamb in a bucket for weighing .......................... 13
Picture 3. Older lamb being weighed with a scale under the floor of the cage ..... 15
Picture 4. The Uno R ATmega328 single-chip microcontroller ......... 22
Picture 5. The ESUMIC LCD-Screen module .................................. 24
Picture 6. The MicroSD, SD-card reader module .............................. 26
Picture 7. The RC522 RFID module ................................................ 27
Picture 8. MAX3232 RS232 to TTL Converter Module .................... 28
## Terms and Abbreviations

<table>
<thead>
<tr>
<th>Term</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>A-weight</td>
<td>age and gender adjusted weight of a lamb</td>
</tr>
<tr>
<td>A42</td>
<td>A-weight of a 42 day old sheep</td>
</tr>
<tr>
<td>A120</td>
<td>A-weight of a 120 day old sheep</td>
</tr>
<tr>
<td>PWM</td>
<td>Pulse-width modulation or pulse-duration modulation, is a method of reducing the average power delivered by electrical signal</td>
</tr>
<tr>
<td>ICSP</td>
<td>In-circuit serial programming, is the ability of some programmable logic devices to be programmed while installed in a complete system rather than requiring the chip to be programmed prior to installing it into the system.</td>
</tr>
<tr>
<td>RISC</td>
<td>Reduced instruction set computer, is one whose instruction set architecture allows it to have fewer cycles per instruction than complex instruction set computers</td>
</tr>
<tr>
<td>MHz</td>
<td>Megahertz</td>
</tr>
<tr>
<td>GHz</td>
<td>Gigahertz</td>
</tr>
<tr>
<td>C/C++</td>
<td>Names of general purpose programming language</td>
</tr>
<tr>
<td>SD-card</td>
<td>Secure digital card is a small flash memory card</td>
</tr>
<tr>
<td>V</td>
<td>Volt</td>
</tr>
<tr>
<td>GND</td>
<td>Ground</td>
</tr>
<tr>
<td>RST</td>
<td>Reset</td>
</tr>
<tr>
<td>SPI</td>
<td>Serial peripheral interface is a synchronous serial communication interface used for short distance communication.</td>
</tr>
<tr>
<td>MISO</td>
<td>Master in slave out, part of SPI</td>
</tr>
<tr>
<td>Abbreviation</td>
<td>Description</td>
</tr>
<tr>
<td>--------------</td>
<td>-------------</td>
</tr>
<tr>
<td>MOSI</td>
<td>Master out slave in, part of SPI</td>
</tr>
<tr>
<td>SCK</td>
<td>Serial clock, part of SPI</td>
</tr>
<tr>
<td>SS</td>
<td>Slave select, part of SPI</td>
</tr>
<tr>
<td>I²C</td>
<td>Inter-Integrated Circuit, is synchronous serial communication bus. Used for attaching lower speed peripheral ICs to processors and microcontrollers in short distance communication.</td>
</tr>
<tr>
<td>SDA</td>
<td>Serial data line, part of I²C</td>
</tr>
<tr>
<td>SRAM</td>
<td>Static random-access memory</td>
</tr>
<tr>
<td>EEPROM</td>
<td>Electrically erasable programmable read-only memory</td>
</tr>
<tr>
<td>AC</td>
<td>Alternating current</td>
</tr>
<tr>
<td>DC</td>
<td>Direct current</td>
</tr>
<tr>
<td>mA</td>
<td>Milliampere</td>
</tr>
<tr>
<td>RX</td>
<td>Receive</td>
</tr>
<tr>
<td>TX</td>
<td>Transmit</td>
</tr>
<tr>
<td>kBd</td>
<td>Kilobaud</td>
</tr>
<tr>
<td>H/L</td>
<td>High/ Low</td>
</tr>
<tr>
<td>LCD</td>
<td>Liquid crystal display</td>
</tr>
<tr>
<td>IRQ</td>
<td>Interrupt request</td>
</tr>
<tr>
<td>nm</td>
<td>Nanometer</td>
</tr>
<tr>
<td>RF</td>
<td>Radio frequency</td>
</tr>
<tr>
<td>RTC</td>
<td>Real time clock</td>
</tr>
</tbody>
</table>
TTL Transistor–transistor logic, it signifies that transistors perform both logic functions and the amplifying function
1 Introduction

The main goal of this thesis was to design a machine, which could collect information automatically from RFID ear tags and the weight data from a scale. The machine would use LCD-screen to show the current progress of the weighing and give instruction on the use. Finally the data would be stored on a micro SD-card from where it can be transferred to a computer.

This thesis will introduce the theory of tracking the growth of sheep and the process of building the machine with electrical and software planning included. The work is done mostly as an independent project work, with some guidance from the commissioner. The finished prototype will be given to the commissioner so that they can design a case and a circuit board for the machine.

The programming is done with the Arduino IDE. Arduino IDE is an open-source Arduino software. It makes it easy to write code and upload it to Arduino based boards. It runs on Windows, Mac OS X and Linux. The environment of the software is written in Java and it is based on open-source software. In picture 1 there is the representation of the main window of the Arduino IDE. (Arduino 2019)

![Arduino IDE](image)

**Picture 1. Arduino IDE**
2 Goal and background of the Thesis

Mietalan Lammastila is a small family owned farmyard owned by Ismo Mietala. He and his wife Maarit Mietala decided to start raising sheep full time in 2014, with only 5 sheep in the beginning. The number of sheep has grown to over 300 in a few years.

Currently the Mietalas are taking part in a project to automate their daily work and to lower the required amount of time and resources needed to take care of the animals. The project already includes automated food and water supply for the animals and many other things that are coming in the future. This thesis is the last part of the automation project and its outcome will eventually be integrated with the other automation systems.

Tracking the weight of sheep must be completed several times during the life cycle of the animal. This is done to keep track of the animal growth and to estimate the potential selling dates. Thus weight tracking plays a vital part in estimating the future profits of the farm. Weight tracking is a time consuming process as many farms in Finland have, on average, only a few workers and in 2016 the average number of sheep each farm had was around seventy. (Suomen Lammasyhdistys, 2019).

Nowadays you can find several different solutions to these problems, but usually their price can vary from hundreds of euros to thousands. As the average age of farmers in Finland is around 50 years, most of them are slow to update to new and expensive systems. (Mela.fi, 2017) Small farms only have small profit margins and these kind of investments are most of the time beyond what can reasonably be expected, so cheap and easy to use solutions are greatly welcomed in the industry.

This thesis seeks to solve these problems by designing a machine that will reduce the amount of paperwork, lessen the amount of workers needed to complete all necessary steps and make it possible to do it in less time. The machine will also be designed so that it will be easy for anyone to replicate and apply on as many farms as possible. To reach these requirements it was decided to use Arduino based developer boards as the brain of the machine. This will make it possible to create complex and flexible devices the costs of which are under 50€.
3 Theory of Animal Growth Tracking

There are several points in the lambs’ life cycle when weighing needs to be done. This does not mean that it could not be done more often but the points that are clarified in the following chapters are the absolute minimum that are required.

The weighing should be aimed to a time when the lambs are at the same age, but occasionally this is not possible due to time constraints. In these situations, it is recommended to convert the weight of all lambs to the same day weight afterwards using the average daily growth values. For example, if you weigh most of the lambs when they are six week old but have to leave some for later, you can use the daily growth values to convert the weight back to what it was at the age of six weeks. This will not give you 100% accurate results but it is better than not doing it. (Greta S. 1998. Elämä lampaiden kanssa)

3.1 Birth weight

The first time when lambs should be weighed is most commonly when they are three days old, but if there is not enough time, you can use the value of 200 grams of growth a day to estimate the weight at the age of three days.

The weighing process is usually done at the same time as marking the lambs for the first time. The easiest way to do it at this point, is by placing the lambs in a small container as seen in picture 2 to keep them still and weighing them in it. (Greta S. 1998. Elämä lampaiden kanssa)
The birth weight is important to know, when you are tracking the starting growth of the lambs. Imagine a lamb that is six-week-old and weighs 10 kilograms. If the weight at birth was four kilograms then the growth has been below average, but if the weight at birth was only two kilograms, the growth of this lamb has been above average. When this information is available you can also tell more accurately the capabilities of each sheep to produce young. The birth weight can also tell if the feeding plan during the pregnancy was correctly made and that the animal has had a proper access to the food provided. Low birth weight usually tells that you need to increase the amount of food given to the sheep next year.

If for some reason this weighing is not done, the birth weight of three kilograms is used.

3.2 Six Week weight

The next weighing for the lambs is when the lambs are six weeks old, but it can be done varying from the age of five to seven weeks and then changed to the six week weight using the average daily growth rate so far.

This weighing is important when looking at the sheep’s ability to raise its young. This metric tells the breeding value and the lamb production capabilities of the mother. (Greta S. 1998. Elämä lampaiden kanssa)
To find the best quality breeding sheep you should look for the ability to give birth to many lively and strong lambs, the capability to take care of the lambs and the ability to produce enough milk for all of them. These requirements are, of course, only valid if the mother sheep has done all the work by itself. Otherwise the farmer must use his own judgement and this weighing loses its importance. (Greta S. 1998. Elämä lampaiden kanssa)

After weighing it is important to react immediately if some lambs are not growing as well as others. At this point it is usually easy to fix, especially if some lambs from the same sheep grow well and others do not. Then you can just separate the larger ones to their own pen, so the mothers can focus on taking care of the smaller ones. (Greta S. 1998. Elämä lampaiden kanssa)

### 3.3 Four Month weight.

The final mandatory weighting for the lambs is when the lambs are 4 months old but it can be done varying from the age of three to five months and then changed to the four-month weight using the average daily growth rate so far. From here the weighting can be done using a small cage with a scale under the floor as seen in picture 3.
This weighing is the best measurement of the lamb’s own growth capabilities and helpful for finding the sheep that produce the largest amount of lambs that have great growth potential. The lambs of these sheep are the optimal selection to keep for future breeding. This is also the time to pay close attention if some lambs are not growing as well as others. Low growth at this age is usually caused by parasites or other diseases. (Greta S. 1998. Elämä lampaiden kanssa)

3.4 Other weighings

It is good to keep weighing the sheep around every 2-3 weeks after the 4 month weighing as this way it is possible to tell more accurately at which stage the growth of each lamb is and when it should be sent to the butchers. The last weighing should be done just before the lamb is sent away. (Greta S. 1998. Elämä lampaiden kanssa)
This kind of repeated weighing is useful as it prevents the farmer from sending underweight lambs to the butchers and prevents the overfeeding of the lambs which are already at the correct weight. This will have a great financial impact to the farm.

3.5 Using the results

The results from the weighting are normally so varied that as such they are not a good source of information. Because of this, the farmer must first change the information to a usable form. Table 1 has an example results of two lambs.

Table 1. Example weight table

<table>
<thead>
<tr>
<th>Gender</th>
<th>3 days</th>
<th>6 weeks</th>
<th>Age in days</th>
<th>4 months</th>
<th>Age in days</th>
</tr>
</thead>
<tbody>
<tr>
<td>Ewe</td>
<td>3kg</td>
<td>15kg</td>
<td>35</td>
<td>30kg</td>
<td>130</td>
</tr>
<tr>
<td>Ram</td>
<td>4kg</td>
<td>20kg</td>
<td>45</td>
<td>40kg</td>
<td>120</td>
</tr>
</tbody>
</table>

In this example both lambs have been weighed at slightly different ages. This is why the weights must be first changed to a more comparable form by using the age. The ewe has grown 12kg in 32 days, so the daily growth value for this lamb is

\[
\frac{m}{t_w} = W \tag{1}
\]

Where “W” is the daily growth value of the lamb, “m” is the weight the lamb has gained after the last weighing in kilograms and “t_w” is the time from the last weighing in days.

This gives the daily growth value of 0.375kg/day. It can be assumed that the lamb continues to grow at about the same rate and turn the slightly early weighing to the 6 week or 42-day weight. We can do this by adding to the measured 15kg weight 7 days of growth with:

\[
((t - t_w) \times W) + W_w = W_n \tag{2}
\]
Where “t” is the wanted age of the lamb in days, “tw” is the age of the lamb when weighed in days, “W” is the daily growth value, “Ww” is the original measured weight of the lamb in kg and “Wn” is the adjusted weight of the lamb in kilograms.

7 days * 0.375kg/day = 2.6kg so the ewe’s weight at 6 weeks would be 17.6kg.

On the other hand, the ram has grown 16kg in 42 days, so by using the equations (1) and (2) we get the following results: the daily growth value is 0.38kg/day and the adjusted weight at 6 weeks is 18.9kg.

Same way the 4-month weight must be changed so that it will be representing the 120-day weight. The calculation can be done with the equation (1) by taking the growth of the lamb between the weightings and dividing it with the time between weighting, the ewe gets a daily growth value of 0.16kg/day between the 6-week weighting and the 4-month weighting. With this the adjusted weight for the ewe is 28.4kg and with the 0.27kg daily growth the adjusted weight stays the same as the one measured for the ram.

If the gender of all the lambs was the same then the farmer could compare them directly to each other, but rams usually grow faster than ewes, so the ewes are given some benefits. This is done by taking the adjusted weight averages from all the ewes and adjusted weights from all the rams and then the difference between these two groups is calculated.

\[ W_{nr} - W_{ne} = W_d \]  \hspace{1cm} (3)

Where “Wne” is the average of the adjusted weights of the ewes and the “Wnr” is the average of the adjusted weights of the rams and “Wd” is the difference between these weights in kilograms.

For example, if in this situation the average adjusted weight of all the ewes was 15.5kg when they were 42 days old and the adjusted weight for the rams was 17kg then the difference is 2kg. Half of this value is added to the ewes’ weight and the other half is subtracted from the rams’ weight.

\[ W_{ne} + (W_d/2) = W_e \]  \hspace{1cm} (4)
\[ W_{nr} - \left( \frac{W_d}{2} \right) = W_r \] 

(5)

By doing this the weights can be compared even when lambs are of different genders. The age and gender adjusted results for the example lambs would be 16kg for ewes and rams after 42 days of growth. This weight is called the lamb’s A-weigh.

### 3.6 Uses of A-weight

With the A-weights the farmer can calculate an index value for all sheep and lambs. This gives one an easily readable value, which evens out the differences caused by the environmental aspects of the lamb’s life and can be used to determine the growth of the lamb and the mother qualities of the sheep. This index value is one of the values farmers look when they are deciding which lambs to keep for future breeding.

The information from the weighings should be collected to a table at this point. There are many ways to present everything in a readable format but the following list of columns is a good way to organise the information so that it include most of the information needed, example of this can be seen in table 2:

1. Tag numbers of the lamb.
2. Age and gender adjusted weight at the age of 42 days (A42).
3. Age and gender adjusted weight at the age of 120 days (A120).
4. Growth of the lamb between the two weighings (A120-A42).
5. Difference between the lamb’s growth and the average lamb’s growth (\( A_d \)).
6. Growth index value of the lamb.
7. Mothers’ name or tag number.
Table 2. Example table of data for choosing the breeding sheep

<table>
<thead>
<tr>
<th>Lamb</th>
<th>A42</th>
<th>A120</th>
<th>A120-42</th>
<th>Ad</th>
<th>Index</th>
<th>Mother</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>16.5</td>
<td>31.2</td>
<td>14.7</td>
<td>-2.7</td>
<td>97.3</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>17.4</td>
<td>37.1</td>
<td>19.7</td>
<td>2.3</td>
<td>102.3</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>15.0</td>
<td>29.3</td>
<td>14.3</td>
<td>-3.1</td>
<td>96.9</td>
<td>A</td>
</tr>
<tr>
<td>4</td>
<td>18.5</td>
<td>40.7</td>
<td>22.2</td>
<td>4.8</td>
<td>104.8</td>
<td>B</td>
</tr>
<tr>
<td>5</td>
<td>13.4</td>
<td>26.6</td>
<td>13.2</td>
<td>-4.2</td>
<td>95.8</td>
<td>B</td>
</tr>
<tr>
<td>6</td>
<td>15.0</td>
<td>30.9</td>
<td>15.9</td>
<td>-1.5</td>
<td>98.5</td>
<td>C</td>
</tr>
<tr>
<td>7</td>
<td>21.7</td>
<td>43.8</td>
<td>22.1</td>
<td>4.7</td>
<td>104.7</td>
<td>C</td>
</tr>
</tbody>
</table>

The $Ad$ value is calculated with the equation (6) and the index with the equation (7).

$$L_g - A_g = A_d \quad (6)$$

Where the “$L_g$” is A120-A42 value of single lamb and “$A_g$” is the average A120-A42 value of all lambs.

$$A_d + 100 = I_g \quad (7)$$

Where the “$I_g$” is the lambs growth index.

The high index values tells the farmer that those lambs have the greatest growth ability. From this the farmer can also tell which sheep has the best qualities for raising lambs by calculating the lamb product and index values for each sheep.

The lamb product value for the sheep is calculated by taking all the labs the sheep has given birth to and adding all of the A120-A42 weights together, in example for sheep “A”, it would be $15.7 + 19.7 + 14.3 = 49.7$, for sheep “B”, it is $18.5 + 13.4 = 31.9$ and for sheep “C”, it is $15.0 + 21.7 = 36.7$

The lamb product values can be used with equation (8) to calculate the index values for each sheep. The more the sheep index value is over the average of 100 the better the production value of that sheep is.

$$L_p - A_{lp} + 100 = I \quad (8)$$
Where the “$L_p$” is the lamb production value for one sheep, “$A_{Lp}$” is the average lamb production value from all sheep and the “$I$” is the index value for the sheep.

Table 3. Sheep index values

<table>
<thead>
<tr>
<th>Sheep</th>
<th>$L_p$</th>
<th>Index</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>49.7</td>
<td>110.3</td>
</tr>
<tr>
<td>B</td>
<td>31.9</td>
<td>92.5</td>
</tr>
<tr>
<td>C</td>
<td>36.7</td>
<td>97.3</td>
</tr>
</tbody>
</table>
4 Hardware

4.1 Selecting the parts to be used

The project was started by making a general list of the parts needed. This included a microprocessor, an LCD-Screen, a micro SD-card reader, an RFID-reader, an rs232, a few push buttons, wires and other accessories.

All the components needed for this project were purchased from several different retailers and they were not the only viable components. A decision was made to use ATmega328 based microprocessors since they are the most popular among the hobbyist groups, information for programming, how to use and different error states are easily available. This makes troubleshooting problems easier for someone with not much experience in microcontrollers.

4.2 Uno R3 Microprocessor

The Uno R3 is a microcontroller board is seen at the picture 4 is based on the ATmega328 single-chip microcontroller that has an 8-bit RISC processor core. It has 14 digital input/output pins, six of which can be used as PWM outputs, six analog inputs, a 16 MHz crystal oscillator, an ICSP header, and a reset button. Programming is done by connecting the board to a computer with a USB cable. The board can be powered with the USB cable, an AC/DC adapter or through a battery. (Arduino Uno.)

Characteristics of the Uno R3 microprocessor:

- operating Voltage 5V
- recommended input Voltage 7-12V
- input Voltage limits 6-20V
- digital I/O Pins 14 (of which 6 provide PWM output)
- analog Input Pins 6 DC
- current per I/O Pin 40 mA DC
- current for 3.3V Pin 50 mA
- flash Memory 32 KB of which 0.5 KB used by bootloader
- SRAM 2 KB
- EEPROM 1 KB
- clock Speed 16 MHz. (Arduino Uno.)

Picture 4. The Uno R ATmega328 single-chip microcontroller.

All the 5v digital and analog pins can be used as input or output although the analog pins can only be used as digital outputs. This is possible with the digitalWrite() and digitalRead() commands in the program. (Arduino Uno.)

Some pins also have special functions in the system:

- Pins 0 and 1 are used as RX and TX pins respectively to read and write serial data.
- Pins 2 and 3 can be used as an external interrupt, meaning they can be programmed to interrupt the program when they receive a specific signal.
- Pins 3, 5, 6, 9, 10 and 11 are PWM pins, meaning they can generate an 8-bit PWM signal with the analogWrite() function.
- Pins 10, 11, 12, 13 support SPI communication using the SPI library.
- Pin 13 has a built in led connected to it. When the pin is HIGH the LED is on and when it is LOW the LED is off. (Arduino Uno.)

This ATmega328 based microprocessor board is used as the brain and the base of the machine. By connecting the wanted modules to the board, it is possible to use all the necessary features needed. The open source developing tool uses C programming language for programming.
4.3 ESP32

Another option for the processor board would have been the ESP32 development board. ESP32 is a single 2.4 GHz Wi-Fi-and-Bluetooth combo chip designed with an ultra-low-power 40 nm technology. It is designed to achieve the best power and RF performance, showing robustness, versatility and reliability in a wide variety of applications and power scenarios. (ESPN32 datasheet)

The ESP32 is capable of functioning reliably in industrial environments, with an operating temperature from -40 to +125 degrees Celsius. Powered by advanced calibration circuitries, the ESP32 can dynamically remove external circuit imperfections and adapt to changes in extreme conditions. (ESP32 overview)

The ESP32 CPU and memory features are following:
- Xtensa® single-/dual-core 32-bit LX6 microprocessor
- 448 KB ROM.
- 520 KB SRAM.
- 16 KB SRAM in RTC
- Support for multiple flash/SRAM chips. (ESPN datasheet)

The ESP32 board would have been also a great option for this project. With its inbuilt Bluetooth transmitter, great sturdy design capable of working even in -40 degrees and large amount of memory it would have been able to perform even better than the ATmega board chosen. The negative point of the board is that it is a lot more expensive than the cheapest atmega328 boards.

4.4 ESUMIC LCD-Screen

The ESUMIC LCD-Screen module that is shown in picture 5 has a wide viewing angle, high contrast and it can display two rows of 16 letters at the same time.

Other features of the ESUMIC LCD-Screen:
- 5 x 8 dots with cursor
- a built-in controller (KS 0066 or Equivalent)
- +5V power supply (Also available for +3V)
- 1/16 duty cycle, back light to be driven by pin 1, pin 2 or pin 15, pin 16 (Vishay)

Table 4. LCD screen pins

<table>
<thead>
<tr>
<th>Pin</th>
<th>Symbol</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Vss</td>
<td>GND</td>
</tr>
<tr>
<td>2</td>
<td>Vdd</td>
<td>+3 or +5V</td>
</tr>
<tr>
<td>3</td>
<td>V0</td>
<td>Contrast Adjustment</td>
</tr>
<tr>
<td>4</td>
<td>RS</td>
<td>H/L Register Select Signal</td>
</tr>
<tr>
<td>5</td>
<td>R/W</td>
<td>H/L Read/Write Signal</td>
</tr>
<tr>
<td>6</td>
<td>E</td>
<td>H → L Enable Signal</td>
</tr>
<tr>
<td>7</td>
<td>DB0</td>
<td>H → L Enable Signal</td>
</tr>
<tr>
<td>8</td>
<td>DB1</td>
<td>H/L Data Bus Line</td>
</tr>
<tr>
<td>9</td>
<td>DB2</td>
<td>H/L Data Bus Line</td>
</tr>
<tr>
<td>10</td>
<td>DB3</td>
<td>H/L Data Bus Line</td>
</tr>
<tr>
<td>11</td>
<td>DB4</td>
<td>H/L Data Bus Line</td>
</tr>
<tr>
<td>12</td>
<td>DB5</td>
<td>H/L Data Bus Line</td>
</tr>
<tr>
<td>13</td>
<td>DB6</td>
<td>H/L Data Bus Line</td>
</tr>
<tr>
<td>14</td>
<td>DB7</td>
<td>H/L Data Bus Line</td>
</tr>
<tr>
<td>15</td>
<td>A/Vee</td>
<td>+4.2V for LED/Negative Voltage Output</td>
</tr>
<tr>
<td>16</td>
<td>K</td>
<td>Power Supply for B/L (OV)</td>
</tr>
</tbody>
</table>

![Picture 5. The ESUMIC LCD-Screen module.](image)
The LCD module is used to display information during the weighting process. The screen will show the tag number, weight and instructions based on the current program mode. The table 4 show the pinouts for the LCD screen.

4.5 OLED Display

The OLED is an advanced display which is self-light emissive and has high contrast, high response time, low power consumptive, and large view angle, with no backlight requirement. Each OLED display is made of 128x64 individual OLEDs, each one is turned on or off by the controller chip. This miniature display can be connected directly to any 3V or 5V microcontroller which has an I²C interface, without needing any kind of level shifter. (OLED Module User Manual.)

OLED displays are very popular with the hobbyist community as they provide greater amount of control about what kind and how much information is shown on the screen, with low amount of input pins. With ability to display large amount of text and pictures on the screen at the same time, the display allows easier access to wanted information, which would make it ideal for a project like this.

The negative of these kinds of screens is that, the libraries the screens use require a lot of memory to employ properly. After some testing on a different project, it was concluded that the screen could not be run with all the other modules at the same time, so the lighter 16x2 LCD screen was chosen instead.

4.6 MicroSD, SD-Card Reader

The MicroSD adapter that is shown in picture 6 is a micro SD-card reader module that can be used to read and write information from a micro SD-card and the SPI interface drive. It supports micro SD-cards and the high-speed micro SDHC-cards. The module needs 4.5V -5.5V or 3.3V power supply and it has six pins that are GND, VCC, MISO, MOSI, SCK and CS.
The SD-card reader is used as a temporary storage for collected information. With the help of the SD-card we can create .txt files that can be read by Excel or similar products, and it will also ensure that no matter how long the machine is in use all the information has been saved.

4.7 RC522 RFID Reader

The RC522 RFID module that is shown in picture 7 has MFRC522 chip-based board, which makes it highly integrated reader/writer for contactless communication at 13.56 MHz possible. The read range is approximately 3 cm and its transfer speed can be up to 10Mbit/s. The module has eight pins SCK, MOSI, MISO, SDA, IRQ, GND, RST and VCC.

Other features of the RC522 RFID reader:

- Highly integrated analog circuitry to demodulate and decode responses
- Typical operating distance in Read/Write mode up to 50 mm depending on the antenna size and tuning
- SPI up to 10 Mbit/s
- Flexible interrupt modes
- Hard reset with low power function
- Power-down by software mode
- Programmable timer
- 2.5 V to 3.3 V power supply
- CRC coprocessor
- Programmable I/O pins
- Internal self-test (MFRC522 datasheet)
The MFRC522 acts as a slave during SPI communication. The SPI clock signal SCK must be generated by the master. Data communication from the master to the slave uses the MOSI line. The MISO line is used to send data from the MFRC522 to the master. Data bytes on both MOSI and MISO lines are sent with the MSB first. Data on both MOSI and MISO lines must be stable on a rising edge of the clock and can be changed on a falling edge. Data is provided by the MFRC522 on the falling clock edge and is stable during the rising clock edge. (MFRC522 datasheet)

The RFID reader is the most important parts of this device as it allows automatic collection of identification information from animal ear tags. This information is connected to the data got from the scale and this way the machine can create a database of identifying numbers and match weights to them.

4.8 EM-18 RFID module

The EM-18 RFID Reader module operating at 125kHz is an inexpensive solution for your RFID based application. The Reader module comes with an on-chip antenna and can be powered up with a 5V power supply. Power-up the module and connect the transmit pin of the module to receive pin of your microcontroller. Show your card within the reading distance and the card number is thrown at the output. (EM-18 RFID module datasheet)

The EM-18 series RFID readers do not differ much from the MFRC522 readers in functions. Differences can mostly be seen in size and appearance so the decision on selecting between these two was mostly based on size and look of the reader and how commonly it was used in other projects.
4.9 MAX3232 RS232 Adapter

The RS232 adapter that is shown in picture 8 is optional part of this build, but it allows the device to be connected with a cable directly to the scale that is in use at the Mietalan Lammastila.

Picture 8. MAX3232 RS232 to TTL Converter Module.

The MAX3232 transceivers have a proprietary low-dropout transmitter output stage enabling a true RS-232 performance from a 3.0V to 5.5V supply with a dual charge pump. The devices require only four small 0.1µF external charge pump capacitors and has 2 receivers and 2 drivers. (Maxim integrated)

Other solutions would have been wireless transmission with Bluetooth, but wired solution was chosen as it is more reliable.
5 Electrical Planning

The machine required much more planning than was thought in the beginning, big problem was that some modules had to share pins to get all the needed features connected to the same board.

![Image of Arduino and LCD screen module.](image)

The LCD-screen has two 5V inputs for supplying power and for the back light, three ground input pins, one of which is to keep pin 5 permanently on low position and 1 voltage input pin 3 where the supply is directed through potentiometer and it is used to control the screen brightness. The pins RS and R/W are used to choose between reading and writing data from rest of the pins and to store it to the modules internal RAM. The module is wired as show in figure 1.
Most of the problems in this build came from wiring and programming the RFID module. It shares three wires with the MicroSD module: SDA, MISO and MOSI. Sharing is normally done by just directly connecting all inputs to corresponding pin in the processor board and control them through software, but this time it was decided that hardware side solution would be easier which lead to adding 3 relay blocks to control where the data is being sent as seen in figure 2.
The RS232 module was the simplest part to wire as seen in figure 3, only needing two inputs besides the power and ground makes it perfect for replacing quickly if any other form of data interface is needed.

![Diagram of Arduino and 4 buttons](image)

Figure 4. Arduino and 4 buttons.

The button inputs are wired as show in figure 4. The microcontroller reads the voltage differences created by the different resistance values between each button. When a button is pressed the microprocessor receives a value between 0 and 1023 based on the voltage input. This is used to differentiate between the buttons in the program.
6 Programming

After the physical connections have been made, the Arduino must be programmed to perform the task. For this the Arduino’s own IDE-program is used. The IDE can compile, transfer and store the code directly to the Arduino.

The program has been divided in several different functions that all ask input from the user one way or another, in some parts just pressing some buttons is enough to proceed to the next step, others require more complex actions like operating the weighting scale. The loop function only starts the mode selection program and from there the program jumps around based on user inputs.

Figure 5. Programs logic loop
Figure 5 represents the flow of the program and shows the functions the program uses and how they move from one function to the other. There are 10 functions with varying complexities. Some are there just to give options for using the machine and rest are integral for the machine to function and to perform its work.

Functions in the program:
- Setup
- Date select
- Loop
- Mode select
- Read and write data
- Manual and automatic reading of tags
- Collecting the weight data from the animal

6.1 Setup

In setup the program makes sure that everything is working and ready for the rest of the program to function. The setup function initializes the RFID, SD-card and LCD modules and sets the serial communication rate that the computer uses to download programs to the board. This is one of two functions that is only run once after the machine is powered on and requires the machine to be powered off and back on again if some module fails.

6.2 Date select

The date select is second of the two functions that only run once. This function is designed so that when the device is powered on the program asks for the current date from the user. This can be skipped if wanted and in that case the program defaults to date 1.1.2019. The date is the then written on a text file that is stored in a SD-card, automatically dividing each weighting in its own block. The last part of this function, names two columns in the text file under the current date. Column one is named “Tag#” and column two “Weight”.

6.3 Loop

This function is the proper starting point of the program. The program is set up so that the loop function only calls the “mode select” function. After that it can branch out based on the provided inputs.

6.4 Mode select

The mode select function tells the machine which branch of the program is needed. There are two options: “Read tags” or “Write tags”, and based on the selection the program will progress to the appropriate step. This function was made to accommodate the possibility to write own information on RFID tags, so that the reuse or purchase of empty tags would be viable.

6.5 Read and write data

The “write tags” function is used to write any wanted number between 0 and 9999 to a specific block inside the RFID tag. The program first asks for the number wanted and when the tag is brought close enough to the RFID reader module, it writes the number on the tag. This function can be run several times in a row in case a large number of tags needs to be written.

The Read data function, like the mode select function, just asks for an input for deciding how the data from the RFID tag is read. The two choices are “Manual read” and “Automatic read”. This function was added in case the RFID module breaks.

6.6 Manual and automatic reading of tags

The manual read function is meant to be used if the RFID module or tags are not functioning. The function allows the user to write the tag number manually by inputting the correct numbers to the machine.
The automatic read function is used to read the information stored in the animal's RFID tag. By placing the tag near the machine's RFID module the module collects the information stored in a specific area of the tag's storage.

6.7 Collecting the weight data from the animals

The last function is for collecting data from the scale. This function waits for a signal to start data collection from the scale. The scale sends the information in ASCII format in 3 parts. First kilos, then grams, and lastly the scale sends a number zero to tell that everything was transferred successfully. For example, if the animal weighs 43.8kg, the first part of the information contains the number 43, second part contains the number 8 and lastly the scale sends the zero. This information is then processed to one variable and it can be stored to the text file.

6.8 Testing

The testing of the finished product was done with simulated animals and tags. The machine was used to first write numbers on some empty RFID tags, which were then read back one by one while also collecting the weight data from the scale. The scale was used by placing heavy objects on top of it and the results were confirmed from the screen attached to the scale. The output of this whole process is a text file that contains a list of all the animals with their tag number and the current weight.
7 Conclusion

The goal of the thesis was to create a prototype of a machine that can replace the old pen and paper method of collecting information in the field. The work was successful, and the goal was reached.

During the project many changes were made in the functionality, use and scope of the machine, when it came apparent how versatile a platform the Arduino based board is. This both increased the complicity of the program in some points and decreased it in others.

The biggest problem was the programming the board, especially when getting the RFID and SD-card modules to work together. For this I was not able to find a solution but with the customer’s permission I was allowed to explore options outside the software environment and used a hardware solutions instead. There was also one big limitation in the project, which was that the RFID-tags require a passcode. So it is not possible to read or write anything from the tags if you do not know this passcode. Then the RFID-reader in the machine cannot be used with the tag.

Testing the machine was done indoors with empty RFID-tags while attached to the weight scale. I wrote some random numbers on the tags myself so I could test both the reading and writing parts of the RFID module.

Before the machine can be taken into operation, it still needs a proper circuit board, where all the modules can be attached directly to the development board, and a proper casing around all the components. The future development of the device was left for the commissioner as was agreed on the start of the project.
BIBLIOGRAPHY

16 x 2 LCD module datasheet. Available at: https://www.engineersgarage.com/sites/default/files/LCD%2016x2.pdf


Arduino homepage (2019). Available at: https://www.arduino.cc/

Arduino Uno datasheet. Available at: https://www.farnell.com/datasheets/1682209.pdf


ESP32 overview. Available at: https://www.espressif.com/en/products/hardware/esp32/overview


Mela, Maatalousyrittäjien eläkelaitos (2019), Vanhuuseläke. Available at: https://www.mela.fi/fi/elakevakuutus/tyoelakkeet/vanhuuselake


SSD1306 datasheet. Available at: https://cdn-shop.adafruit.com/datasheets/SSD1306.pdf