Gezamenlijke opleiding UHasselt en KU Leuven

2017•2018

Faculteit Industriële ingenieurswetenschappen

master in de industriële wetenschappen: elektronica-ICT

Masterthesis

Board design and implementation for 8-bit avr microcontrollers in an educational environment

PROMOTOR :

Prof. dr. ir. Ronald THOELEN

PROMOTOR :

Mr. Juha SARKULA

Gideon van Roon

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

2017•2018

Faculteit Industriële ingenieurswetenschappen

master in de industriële wetenschappen: elektronica-ICT

Masterthesis

Board design and implementation for 8-bit avr microcontrollers in an educational environment

PROMOTOR :

Prof. dr. ir. Ronald THOELEN

PROMOTOR :

Mr. Juha SARKULA

Gideon van Roon

Scriptie ingediend tot het behalen van de graad van master in de industriële wetenschappen: elektronica-ICT

UHASSELT

KU LEUVEN

Electrical and Automation Engineering
Häme University of Applied Sciences Valkeakoski

Author Gideon van Roon **Year** 2018

Subject Board design and implementation for 8-bit AVR microcontrollers in an educational environment

Supervisor(s) Juha Sarkula

ABSTRACT

This paper is presenting the design process and implementation for a microcontroller board to be used in HAMK's embedded system design course. The microprocessor on the board is an 8-bit AVR ATmega32U4. This controller is capable of driving and reading several onboard peripherals, such as: LED's, an external display, temperature sensor, additional memory, buttons, IR transmitter and IR receiver. The board features an external high voltage power input for use in PLC cabinets. Programming is done with ATMEL Studio without the need for a special programmer.
A development board is realised, meeting the basic requirements. Buttons can be read, LED's can be driven. Communication can be instantiated with the I²C components. The external display is controlled over a dedicated header. The power supply has proven to handle voltages up to $40\ V$ and output $3\ A$ at $5\ V$. Shorting the supply can be done up to forty seconds before reaching the regulators thermal limits.
All components are housed on a custom PCB. Most components are in SMD format external connections.
Cost is lowering when larger quantities are ordered; A single unit costs €87,62, ten units €$21,46$ and the price for one hundred units drops to €15,98.

Keywords 8-bit AVR, development board, Embedded systems

Pages 59 pages including appendices 9 pages

CONTENTS

1 INTRODUCTION

Atmel, part of microchip, produces a variety of microprocessors and microcontrollers using their AVR architecture. The controllers are very popular due to the ease of use and implementation by Arduino on their open-source prototyping boards.

The processing heart of the PCB is an ATmega32U4. The controller can read six buttons and drive eight on-board LEDs. The custom PCB is able to house an I^2C temperature sensor and EEPROM memory. LCD displays can be used in combination with the dedicated header.

The goal is to teach students automation engineering at Häme University of Applied Sciences Valkeakoski the basics of microcontrollers.

2 OBJECTIVE

The objective is to design an 8-bit AVR development board. The board must have the following on-board peripherals: Buttons, LEDs, adjustable voltage output and an I²C temperature sensor. The board must be capable of being powered over USB and over an external power input. External voltages can be as high as $24\ V$. A small formfactor on a custom PCB must be maintained while keeping costs low.
Programming capabilities must include programming over USB within Atmel Studio.

3 COMPONENTS

The components are picked to obtain a low overall cost while maintaining quality and functionality.
A purpose specific printed circuit board (PCB) is designed to accommodate the picked components. The use of a PCB is necessary as some components are available in surface mounted device (SMD) packages only. An advantage is the ability to have a small and organized footprint.

3.1 Microcontroller

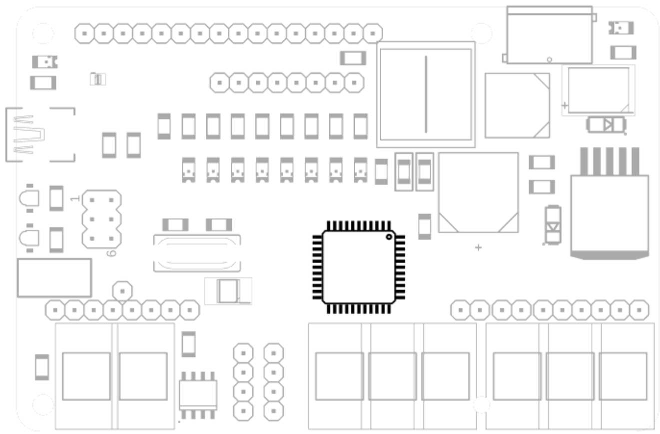

Figure 1: Microcontroller on PCB

An ATmega32U4 is used as processor (highlighted on Figure 1). Comparable PIC controllers have a steeper learning curve, making them less compelling for use in introduction courses. Chips based on Atmel's proprietary AVR-architecture are easier to use. Many examples and projects are already made by the online community. The programming environment of choice is Microchip's own Atmel Studio.
A comparison concluded four different setups for programming. First, a processor with integrated universal serial bus (USB) controller. A second option is picking a separate USB-controller and processor. Next option is a firmware-only implementation with the USB protocol within the bootloader portion of the memory. At last the purchase of a dedicated microcontroller programmer. (Objective Development Software GmbH 2012)
To maintain compatibility with the programming environment within Atmel Studio, a small selection of controllers can be used to program the MCU. This also forms a problem for chips using a programmed USB protocol.

Atmel's AT90USBXXX range microcontrollers are low cost chips. Disadvantages appear when comparing features. The AT90USBXXX microcontrollers do not support the inter-integrated circuit (I^2C) protocol and the amount of GPIO ports is less than those of a similar priced ATmega's.
ATmega U (USB) series chips are sold with two different model numbers; U2 and U4. U2 series controller come without analog digital converters.
The ATmega32U4 provides a good balance between cost and functionality; an integrated USB controller handles the USB connection while the default Device Firmware Uploader (DFU) bootloader is optimized for programming with Atmel Studio.
A chart comparing different AVR controllers is presented in the appendix. (Microchip Technology 2016)

3.1.1 Principal

The principle on which AVR is based is the reduced instruction set computer (RISC) architecture. the Microcontrollers use the principal of executing small 8-bit instructions and saving the results in a cache memory. 256 different instructions can be composed with an 8-bit size. Basic instructions consist out of arithmetic and logic (ALU) operations. For example: comparing, addition, subtraction and multiplication of two numbers. Other operations can for example: requesting and saving values in the cache memory, reading and writing the GPIO port registers or communicate to the serial peripheral interface (SPI) module. All blocks in the AVR enhanced RISC are displayed in Figure 2.

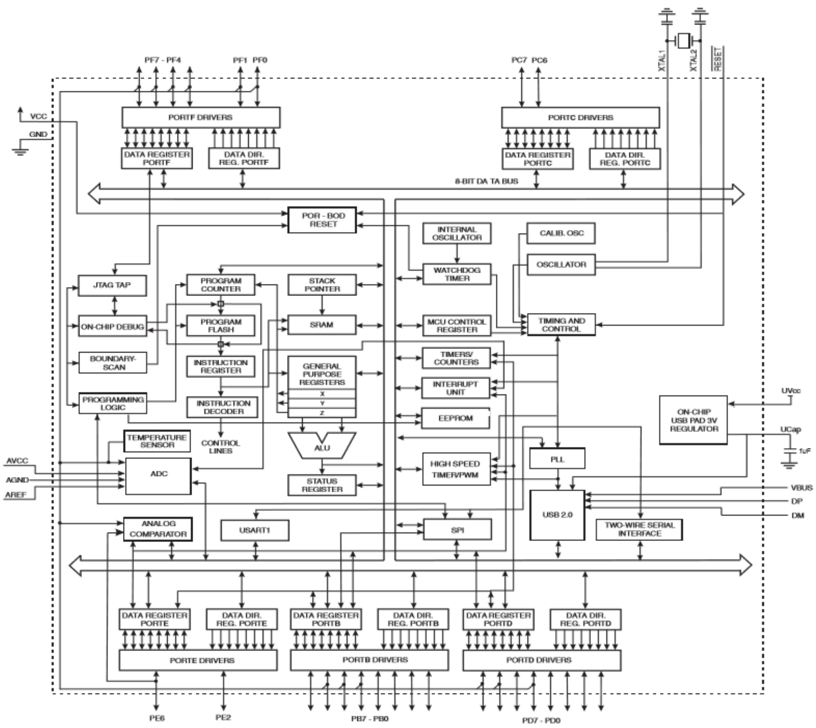

Figure 2: Atemega32U4 block diagram (Microchip Technology 2016)

When the processor is powered, and the bootloader is not triggered, the program counter starts counting at the first instruction. The program counter adds one to the register address it has to read and the next instruction is loaded.
A jump can be executed to execute an instruction on a different register address. For example, in an IF statement; The program counter continues when a condition is true. A jump instruction to a different address is executed when the condition is declared to be false.
Code is converted to bit-level instructions in the compiler. Atmel Studio provides the option to write on instruction level in assembler.
The Architecture is optimized to execute an instruction in one clock cycle.
(Microchip Technology 2016)
(Patterson, D. & Hennessy, J. 2014)

3.1.2 Features

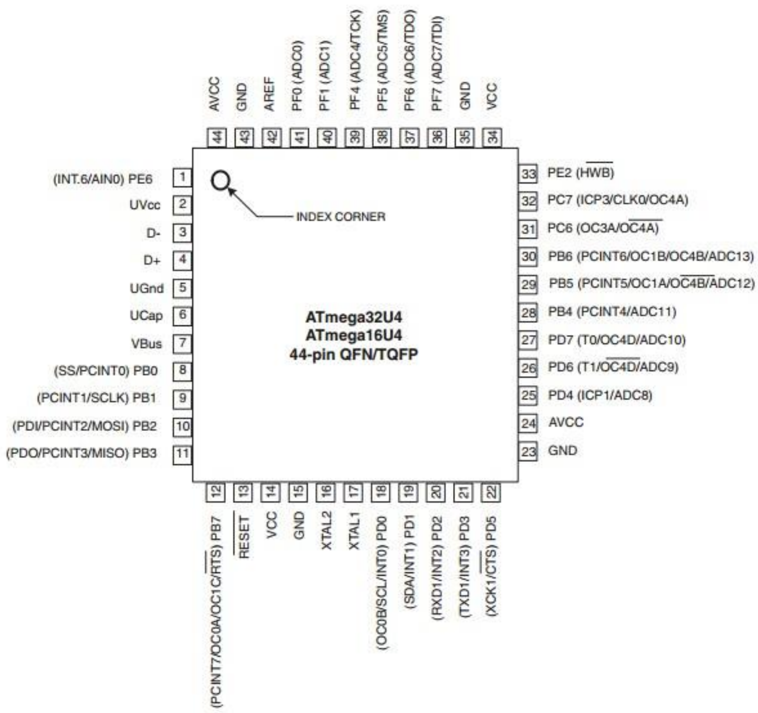

Figure 3. ATmega32U4 TQFP pinout (Microchip Technology 2016)

A wide range of general purpose input and outputs (GPIO) is fitted in the chip. Figure 3 displays the pinout of the 44-pin thin quad flat package (TQFP). Five ports; port B, C, D, E, F are usable as GPIO. Some pins have specific purposes. PB2 and PB3 handle MOSI and MISO. These pins are used when the in-system programmer (ISP) is connected. PD0 and PD1 handle SCK and SCK for I²C communication. PE2 is only available for the HWB button. Ports B, C, D and E can all read and write digital values. PB5,

PB6, PB7, PC6, PC7, PD6 and PD7 double in function as pulse with modulated (PWM) outputs.
The PWM functions by increasing a counter and comparing that counter in a comparator. The interrupt flag on the comparator input (OCnx on Figure 4) resets the counter and flips the output signal. When the counter reaches its maximum value, the counter is reset, and the output value is high again. The earlier the interrupt flag is met, the shorter the duty cycle.

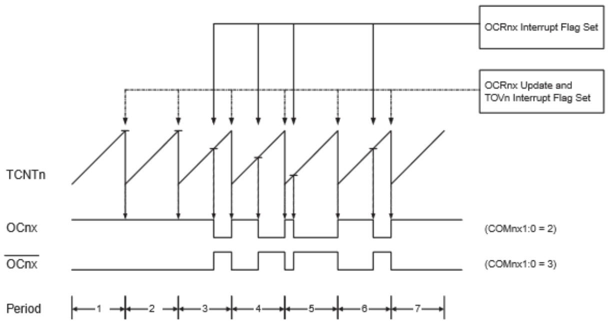

Figure 4: PWM timing diagram (Microchip Technology 2016)

Port F contains only ADCs and is only able to read digital or analog values. (Microchip Technology 2016)

3.1.3 Fuse bits

Fuses contain a variety of advanced settings. The fuses resemble registers that can be set while programming. In order to change the fuse settings, "#include <avr/fuse.h>" must be added to the code.
Settings are divided into three sections; low, high and extended.

```
#include <avr/fuse.h>

FUSES =
{
        .low = (FUSE_CKSEL0 & FUSE_CKSEL1 & FUSE_CKSEL3 &
        FUSE_SUT0 & FUSE_CKDIV8),
        .high = HFUSE_DEFAULT,
        .extended = EFUSE_DEFAULT,
};
```

Online fuse calculators make it easy to select features and set the correct values for the fuses.
(Engbedded 2014)
(Microchip Technology 2016)
(Nongnu 2016)

3.1.4 Bootloader

The memory In the Atmega is divided in two sections; The program memory and the bootloader section. The program memory contains the user uploaded code. The bootloader is the software that runs on the chip to get the code in the program memory. Microchips USB device firmware uploader (DFU) bootloader runs by default on the Atmega USB chips.
Uploading code is using in-system programming from an USB host-controller or the ISP pins on the board. Programming over the external ISP port requires an additional programming interface.
The microcontroller jumps to the programming memory by default. In order to access the bootloader a certain hardware condition must be met. The button “HWB”, connected to PE2, provides this condition. When PE2 is pulled down during boot the bootloader code executes and the USB controller instantiates a connection to the host. The Atmega behaves as a FLASH memory programmer for its own internal memory. A bitstream is then stored in the memory. After a reset, the normal execution is continued.
(Microchip Technology 2008)

3.2 **Peripherals**

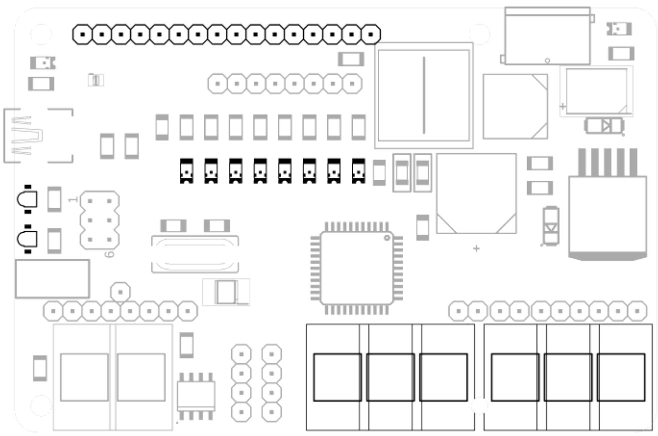

Figure 5. Peripherals on PCB

The development board houses four different peripherals as displayed on Figure 6.
The user can give input with six buttons connected to pins PD2 to PD7. PD0 and PD1 are not used because they double in function for I^2C communication.
Eight light emitting diodes (LED’s) are connected to each of port B’s input/output (IO) pins. The LEDs function as a binary display, capable of

displaying one byte at a time. The least significant bit (PB0) is located on the right.
The 16-pin display header is located on the top left of the board. This header is compatible with Hitachi HD44780 liquid crystal display (LCD) modules, for example used on Arduino's LCD. The basic modules have 2 rows with 16 characters. Each character is composed out of 5x8 pixels. Modules with double the number of rows can be driven over the same header.
(Arduino 2015)
Connected displays are entirely driven by port B. PB0, PB1, PB4 and PB5 populate the 4-bit mode data lines. PB2 is connected to register select (RS) which selects the instruction or data register. PB3 enables the display data inputs. PB6 and PB7 respectively handle the backlighting and contrast. Both ports can provide PWM signals to vary the backlighting and contrast intensity.
Last peripheral is an infrared transmitter and receiver couple. A light emitting diode tuned to output $940\ nm$ light waves is used as transmitter. At the receiving side a npn phototransistor is places where light within $830\ nm$ and $950\ nm$ toggles the base. The set can serve different purposes. For example: communication can be instantiated in-between two boards or can be used as a proximity sensor.
(Vishay 2017)

3.3 GPIO

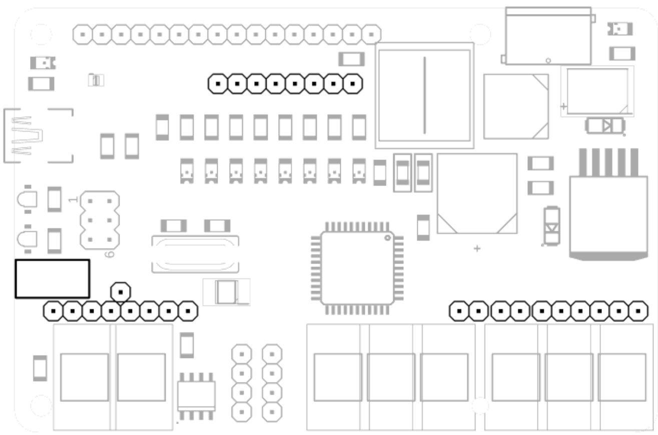

Figure 6. GPIO on PCB

All GPIO ports on the ATmega32U4 are available through female sockets. These ports are on Figure 6:
- Top: PB0 up to PB7
- Bottom right: PC6 and PC7
- Bottom Left: PD0 up to PD7

- Bottom right: PE2 and PE6
- Bottom right: PF0 up to PF7

Connecting to PE2 is available over a female socket but cannot be used. This is due to the HWB function. The microcontrollers fuse settings must be edited to make use of this pin. Enabling this pin as GPIO prohibits programming the Atmega over USB. The processor instructions jump on boot to the program memory. An external programmer is necessary to reprogram the EEPPROM.

A single pin on the board is dedicated to a variable voltage output using a 10k potentiometer. High output accuracy is obtained by the 23 rotations between the minimum and maximum resistance. The potentiometer can handle a maximum of $0{,}5\ W$ over the resistors at 70°C.

(Microchip Technology 2016)

3.4 I²C devices

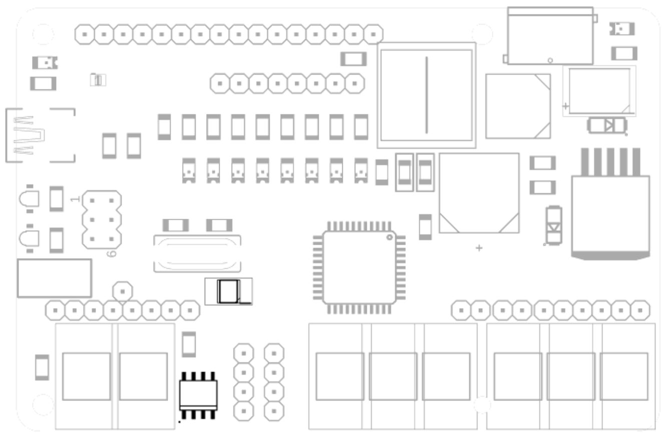

Figure 7. I²C devices on PCB

Two peripherals using the inter-integrated circuit (I²C) protocol can be mounted on the board (bottom on Figure 7); A temperature sensor and an electrically erasable programmable read-only memory (EEPROM) chip. Any other I²C device with a similar pinout and footprint can be used as substitute. Unused chips are not required to be soldered on.

I²C is a bus system, capable of driving multiple devices on a single 2-wire bus. The wires are a data line and a clock line. A master device can instantiate connections. Slave devices may return a value or an acknowledgement over the duplex connection. A pull-up resistor must be placed over the data line. Slaves are only able to pull the line low or release it. The pull-up resistor restores the high state after releasing it.

(Microchip Technology 2016)

3.4.1 Temperature sensor

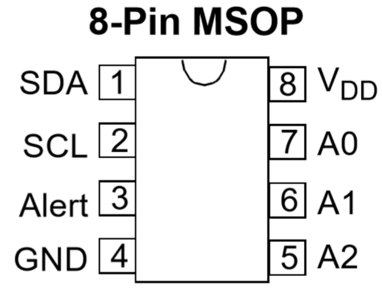

Figure 8. MCP9808 MSOP pinout (Microchip Technology 2011)

Microchips MCP9808 offers best characteristics for a low price. The pinout is dislayed on Figure 8. Measurable temperatures range from -40°C up to 125°C while maintaining a typical accuracy of 0,25°C and a maximum error of 0,5°C. The resolution of the readings can be set to +0.5°C, +0.25°C, +0.125°C and +0.0625°C.
Upper temperature sensing capabilities are limited by the microcontroller and memory IC at +85°C.
The location is chosen to reduce influence by the power supply heat dissipation while maintaining short clock and data lines. Thermal heated areas are displayed on Figure 27.
Reading and writing data to the component uses the I²C protocol. Initiating communication with the sensor is done by contacting its address in the format $0\ 0\ 1\ 1\ A2\ A1\ A0\ R/\overline{W}$. The parameters $A2$, $A1$ and $A0$ are connected to the ground, making the address byte: $0\ 0\ 1\ 1\ 0\ 0\ 0\ R/\overline{W}$. The last $R/\overline{W}$ bit commands a read or write operation.
Pin number 3, Alert, is not connected because this feature can be easily implemented in software on the MCU.
(Microchip Technology 2016)

3.4.2 Memory module

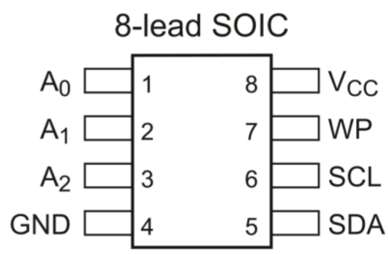

Figure 9. AT24CxxxC SOIC pinout (Microchip Technology 2015)

A SOIC footprint (Figure 9) is located at the bottom for housing additional EEPROM memory modules. The compatible types are Atmel's AT24CxxxC

IC's. Memory capacity ranges from $1kbit$ up to $1Mbit$. An AT24C256C is included in the design and bill of materials. $256kbit$ is large enough for simple datalogging applications. The memory is accessed in rows of bytes, $32\ k$ rows in total.
The contact addresses are checked to be different from the temperature sensor address. Similar formats are maintained except for the prefixes. The memory modules use $1\ 0\ 1\ 0\ A2\ A1\ A0\ R/\overline{W}$ where $A2$, $A1$ and $A0$ are connected to the ground. The address line for any connected module is determined to be $1\ 0\ 1\ 0\ 0\ 0\ 0\ R/\overline{W}$.
Write protect pin, pin number 7, is permanently pulled to the ground. Reading and writing to the chip is only instantiated over I²C without the need of pulling the pin low.
(Microchip Technology 2015)

3.5 Power supply

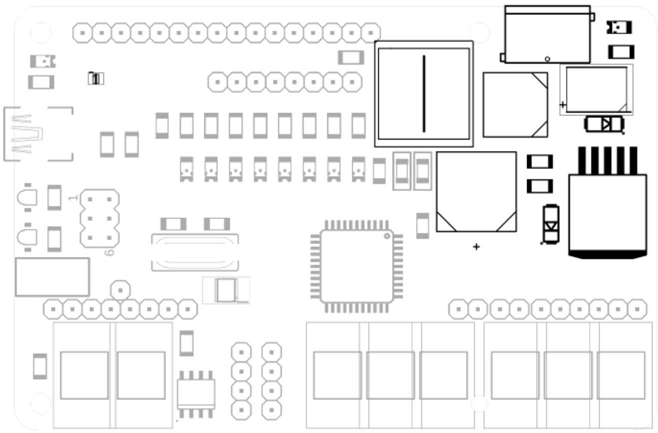

Figure 10. Power supply on PCB

The onboard power supply (in the top right section on Figure 10) is responsible for providing power to the board while it is not connected over USB, for example: while used in PCL cabinets. This also sets the minimal requirement; PLC's mainly use a voltage of twenty-four. Input voltages may vary dependent on the external voltage applied. The regulator steps this down to a steady five volts. A wide range of applications to be powered is made possible by the high current that can be supplied. The maximum rating is two amps or ten watts.
Detailed schematics can be found in the appendix.

3.5.1 Switching regulator LM2596

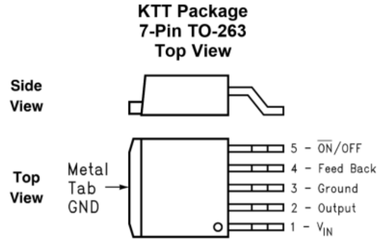

Figure 11. LM2596 TO-263 pinout (Texas Instruments 2016)

A switching regulator has the preference as linear regulators dissipate the entire input to output difference as heat. This comes at a cost as switching regulators are higher in price.
The component of choice is the LM2596 buck, or step-down regulator (Figure 11). These come in a variety of fixed output voltages and adjustable output voltages as well. The components footprint is the same across all different types. This makes it possible to use a fixed $5V$ regulator (LM2596-5.0) or an adjustable regulator (LM2596-ADJ) tuned to $5V$ through a feedback loop. The difference in schematic is only two resistors but the difference in cost is significant. The $5V$ regulators have a single unit price of €$4{,}40$ while the variable regulators start at €1,81. The difference in price justifies for the use of two extra resistors on the board.
The critical circuit conversion consists a single resistor. Differences only exists in the feedback loop. LM2596-5.0 regulators are tuned to 5V output when the output is directly connected to the feedback pin. Regulators of type LM2596-ADJ are required to have a voltage divider in the feedback loop.
The regulator switches at a frequency of $150\ kHz$. To reduce the input voltage, the duty cycle is adjusted. The theoretical duty cycle is calculated using $D[\%] = \frac{V_{out}}{V_{in}} \cdot 100$. The real duty cycle differs slightly due to losses. Simulated duty cycles for $12\ V$, $18\ V$ and $24\ V$ are displayed in Figure 16. The output is controlled with a smoothened signal over the feedback loop.
Necessary methods provide protection against errors made by users. The regulator is picked so that in case a voltage is applied higher than the nominal input. A diode provides protection against inverse polarities.
(Texas Instruments 2016)

3.5.2 Safety measures

Overvolting is prevented by the regulators extended range. Nominal voltages of 24V should be applied but reliability is guaranteed up to 40V.
An input diode of the type SS54 provides protection against inverse a connected polarity. While in reverse bias, the diode can handle up to $40V$.
A second diode of the type SS12 doubles in function. It prevents an applied current on the regulators output while the board is connected over USB.

Overcurrent protection is also provided by this diode. The maximum rated forward current is as much as $2A$ but can be increased by replacement with another type. Documented current ratings for this diode must be lower than the maximum current the switching regulator can handle. The cost for a SS12 diode is as low as €0,07 while a broken regulator is more expensive to replace.
(Multicomp 2011)

3.5.3 Power source selector

The board has two power inputs; over USB and the external terminal connection. A P-channel MOSFET prevents that both sources interfere with each other. USB power is disconnected from the main power line V_{cc} when power is applied to the V_{EXT} terminal.
Threshold voltage V_{gs} between the gate and source must be higher than the microcontrollers minimum operating voltage to ensure continuous operation during source switching.
High source-drain currents flow in the event of an accidental short circuit. To remain a single point of failure, the source-drain current must be rated higher than the fuse diode.
SI2307BDS-T1-E3 from Vishay is in compliance with all requirements. The continuous current is rated at $2.5\ A$ and the threshold voltage at $3\ V$.
(Vishay 2008)

3.5.4 Output signal

Output constancy is visualised using the oscilloscope.
The frequency of the regulator is around $150\ kHz$. Dependant on the input voltage, the generated output on the regulator resembles a triangular or a sawtooth signal. This output is made visible on Figure 12 with a 24V input. The output is smoothed using an inductor and capacitor.
(Texas Instruments 2016)

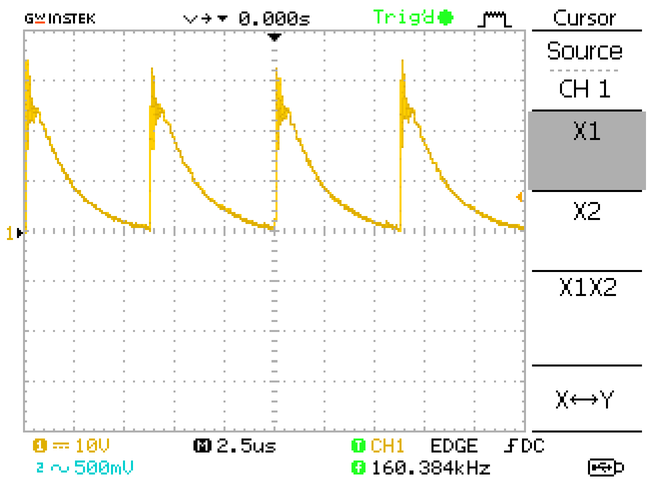

Figure 12. Unsmoothed output signal

Figure 13 displays output wave after applying an inductor and capacitor. $4{,}90\ V$ is read on the output as expected from the LM2596-ADJ. Small noise fluctuations exist on the signal.
The input voltage for Figure 13 is $24\ V$. Similar results are measured with 12V on the input.

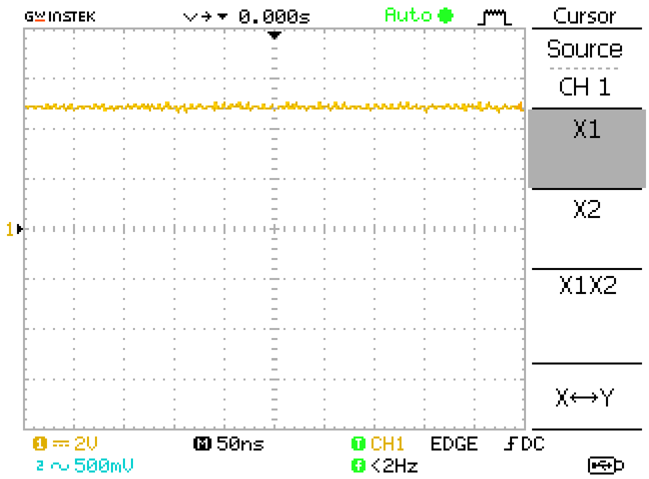

Figure 13. LM2596-ADJ Output

The minimum operating voltage for the microcontroller is $2{,}7\ V$. Minimal input requirements for running the Atmega are met when applying $3{,}3\ V$ on the regulator input. Tuning on the regulators feedback loop is specified for an $4{,}91\ V$ output. An unstable output signal is generated when the input voltage drops to low. Figure 14 shows the unstable $2{,}7\ V$ output signal.

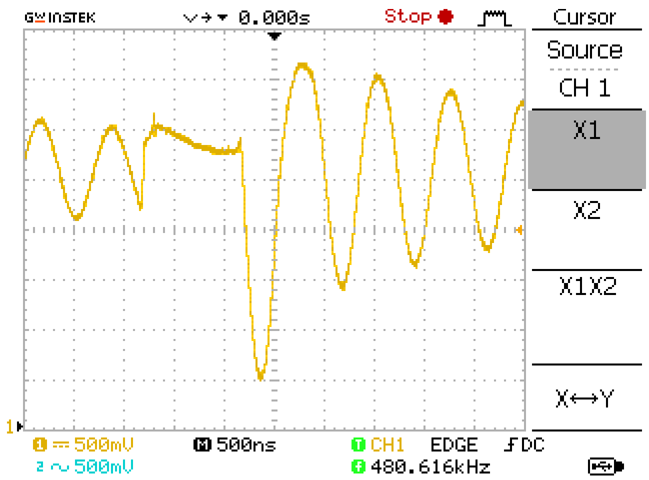

Figure 14. Unstable $2{,}7\ V$ output voltage

To insure a stable output, input voltages must be kept above $6\ V$.

3.5.5 Efficiency

Efficiency and duty cycle curves are plotted using Texas Instruments WEBENCH® Design Center. Only the LM2596-ADJ has been simulated. This component attracts the focus because this is the component that is supposed to be used on the PCB.

The efficiency and its corresponding duty cycle are plotted in Figure 15 and Figure 16. Both simulations consist out of three input voltages: $12{,}0\ V$, $18{,}0\ V$ and $24{,}0\ V$.

Texas Instruments (n.d.)

Two conclusions are drawn regarding the efficiency. First, efficiency rises when the current through the regulator increases. Second, lower input voltages result in a higher efficiency. This can be explained by a larger voltage over the internal resistance, causing more energy to be dissipated. The rise in efficiency is high at a lower current flow but less than 1% in the range from $1\ A$ to $2\ A$.

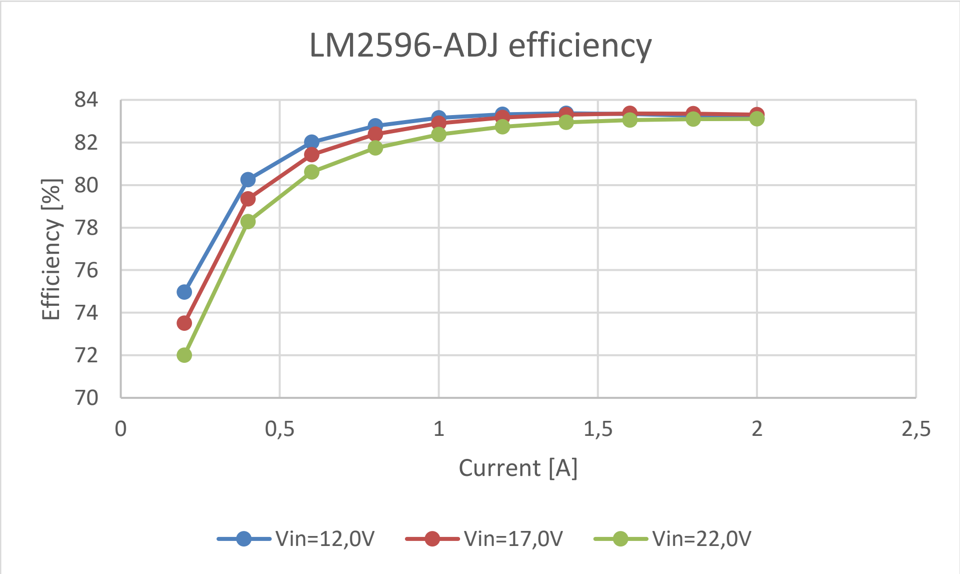

Figure 15. LM2596-ADJ Efficiency

The duty cycles drop when the voltage rises as expected. The largest deviations are found in between the first two data points at $18\ V$ and $24\ V$. The difference is only 2 to 3 percent and is barely noticeable in real-world applications.

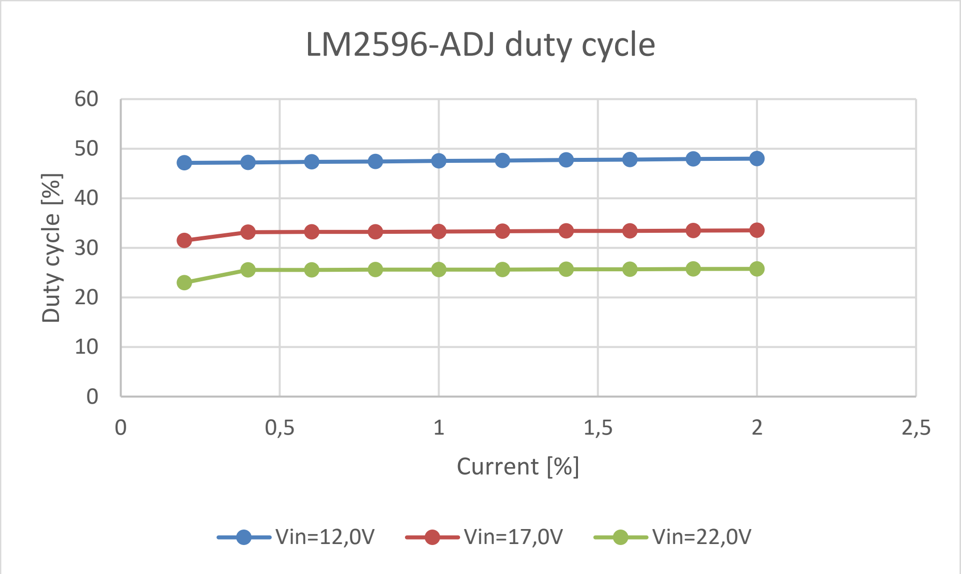

Figure 16. LM2596-ADJ Duty Cycle

4 CIRCUIT

Schematics are designed in EAGLE according to the component datasheets.
A list containing all resistor, capacitor and diode values is found in the appendix
Note that the component names may not be according to the global scheme and parts list.

4.1 Microcontroller

Included in this section is the microcontroller wiring and the circuitry to make it usable.
Figure 17 displays the necessary circuit to make the ATmega32U4 function and usable. This includes the microcontroller itself including several capacitors, a low-pass filter, the USB port, the reset button and the hardware bootloader button. The scheme also displays an input smoothing capacitor and the power on LED.

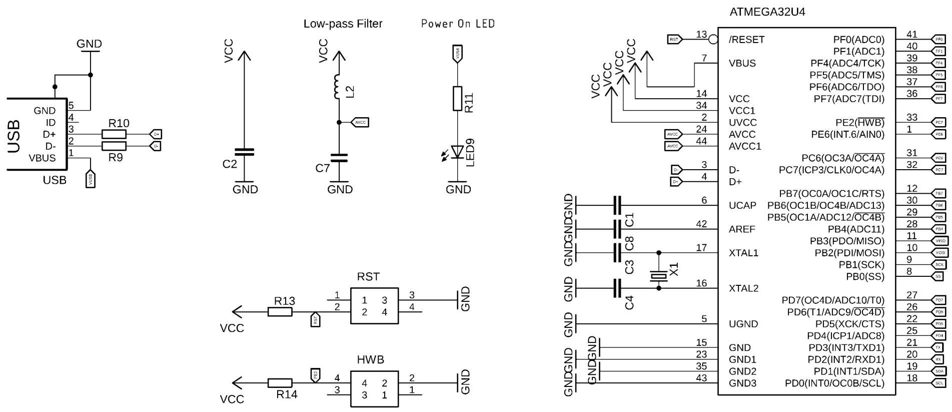

Figure 17. Microcontroller scheme

All necessary connections are displayed on the left in Figure 17. On the right side are IO ports located. HWB can be enabled as GPIO and is located on the right.
Few components are necessary to make the ATmega32U4 function. The amount can be reduced by setting the fuse bits correctly and removing the clock crystal and its capacitors. The oscillator, however, is included in the schematic as there might not be a programmer available to set the fuses.
Both buttons are active high. This prevents the microcontroller to access the bootloader when not pressed and jump to the program instructions. The board continues operation when power restores after a power loss.
AVCC is the power input for the analog circuit inside the chip. An independent power input is fitted to prevent noise from the digital side. The analog circuit is fed through a low-pass filter.

Two 22 Ω resistors are connected to the D+ and D- terminals on the USB port as instructed in the datasheet. The USB lines can be considered as transmission lines. Rule of thumb is that the transmission line should be terminated when the data frequency is above $10\ MHz$. USB 2.0 handles frequencies of $60\ MHz$ and $15\ MHz$ for respectively high speed and low speed. More specific; termination should be applied when $l \geq \frac{t_r}{2 \cdot t_{pr}}$. In this equation is the length presented as $l\ [mm]$, the signal rise time as $t_r\ [ps]$ and the signal propagation speed as $t_{pr}\ [ps/mm]$. For an PCB using FR4 as material, propagation time $150\ ps/inch$ is used. This converts to $5{,}91\ ps/mm$. Rise (and fall) time, t_r, must be lower than $500\ ps$ in 90% of the transitions for high speed USB according to USB specifications. Termination must be applied when the data path length exceeds $l > \frac{500}{2 \cdot 5{,}91}$, $l > 42{,}3\ mm$. Trace lengths on the PCB alone for D- and D+ are respectively $47\ mm$ and $49\ mm$. When Series resistors do not terminate the reflection, they are only reduced. These reflections are caused by differences in impedance between the USB cable, PCB traces and Microcontroller lines.
The USB compliance states that a minimum capacitance of $1\ \mu F$ is required over all USB peripherals. C1 with value $1\ \mu F$ is connected to UCAP for this purpose.
Input capacitor C2 is applied to smoothen the input voltage.
(Jerry Seams 2007)
(Larry Davis n.d.)
(Microchip Technology 2016)
(Portland State University n.d.)
(USB-IF n.d.)

4.2 Peripherals and general-purpose input/output

Four VCC pins and four GND pins are available through headers.
All GPIO pins on all ports, port B, C, D, E and F, are connectable with jumper wires through five female headers. These are displayed in Figure 18.

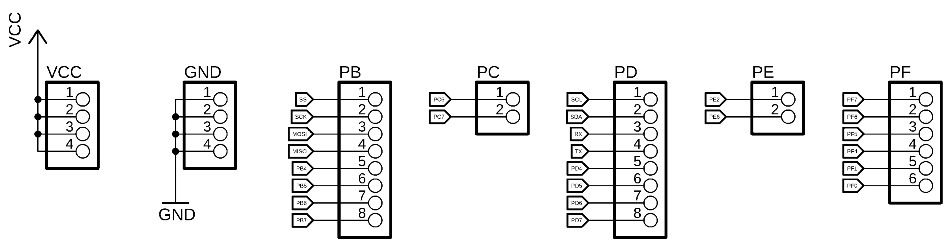

Figure 18. VCC, GND and GPIO ports scheme.

The eight LED's are connected to port B0 up to B7 with a shunt resistor.
Six buttons make use of port D2 up to D7. D1 and D2 are not used because these pins double in function for I²C communication. Pull-up or pull-down resistors are not required. A $10\ k\Omega$ internal pull-up resistor is applied in

the ATmega32U4 when configuring a pin as input. The $10\,k\Omega$ pull-up reduces the current to $0{,}5\ mA$ while active according to ohms law. Pins 1, 2 and 3, 4 are connected when the button is pressed. The LEDs and buttons are displayed in Figure 19. Only a shunt resistor is needed to control the current through the LED. The current through the LED is $20\ mA$ and a voltage of $2{,}2\ V$ over it. Powered with 5V, this means that the shunt resistor catches $2{,}8\ V$ at $20\ mA$. Ohms law provides the resistor value: $140\ \Omega$. The closest E12-series value is $150\ \Omega$. Including a standard $\pm 5\%$ production margin makes this resistor value at least $142{,}5\%$. The chosen $220\ \Omega$ resistor makes the light output on the LED more comfortable to look at. The current is reduced to $13\ mA$ by the shunt resistor.
(Microchip Technology 2016)

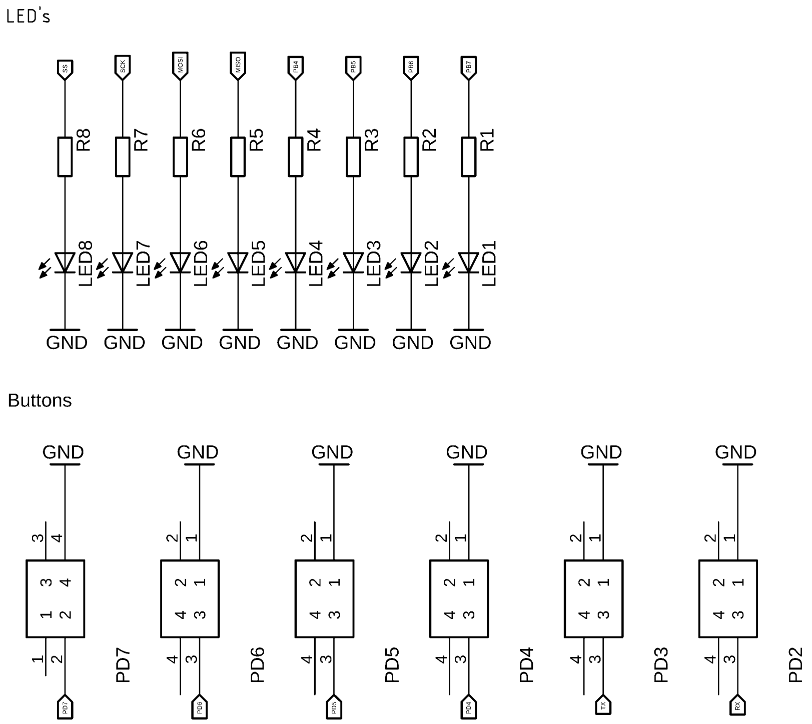

Figure 19. LEDs and buttons scheme

Figure 20 contains an image of the infrared transmitter, infrared receiver and the variable voltage output.
The infrared transmitter is comparable to a regular LED except for the emitted wavelength.
The functioning of the infrared receiver is similar to that of a n-channel transistor. Infrared light caught on the gate switches the transistor on, pulling the input pin on the Atmega to the ground.
(Vishay 2017)

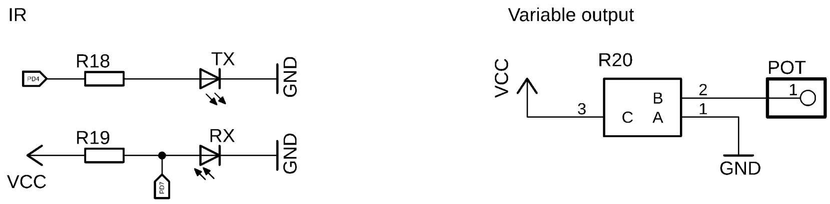

Figure 20. IR transmitter, IR receiver and variable voltage output

A potentiometer is applied as voltage divider to control the output voltage. This output is accessible through a single female header.

4.3 Temperature sensor and memory module

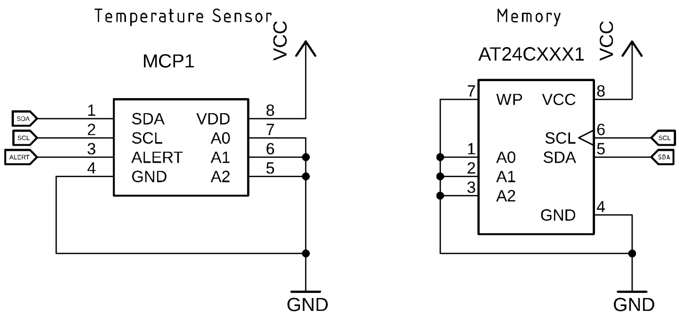

Figure 21. Temperature sensor and memory module scheme

The bus protocol I²C is used to control both IC’s. Both SDA and both SCL pins are connected to the Atmega’s SDA and SCL. The internal pull-up resistors in the Atmega must be enabled on the serial data (SDA) and serial clock (SCL) lines.
A0, A1 and A2 are connected to the ground because the address prefixes differ.
The ALERT pin on the temperature sensor (left on Figure 21) is not connected because this functionality can be implemented on the microcontroller.
Write enable pin (WP) on the AT24 (right on Figure 21) is pulled to the ground, making writing accessible at any time.
(Microchip Technology 2016)

4.4 Power supply

The power supply contains in total twelve components. Terminal, VEXT, serves the purpose of connecting wires. LED1 and R3 provide a visual representation for an external power input. Diodes D1 and D3 protect the regulator against inverse polarities. LM25896-XXX, D2, L1 and C2 ensure the power supply function. A catch diode is required in combination with inductors. An inductor has the property to counter any change in current. Diode D2 prevents the current to flow through the regulator at the falling edge in the regulation. Inductor L1 reduces the ripple in the current. Capacitors C1 and C2 reduce ripple voltages. C1 for the input and C2 for the output.

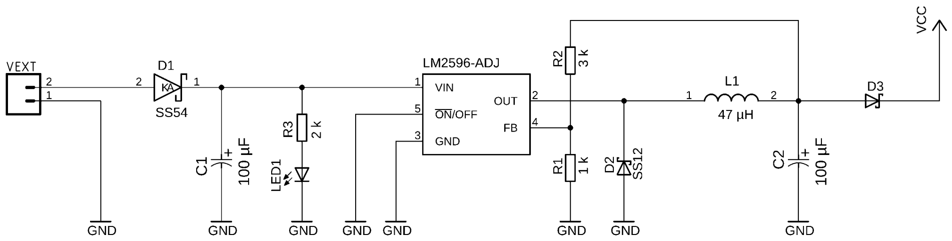

Figure 22. LM2596-ADJ scheme

Figure 22 shows the LM2596-ADJ circuit with a voltage divider in the feedback loop. Formula $V_{OUT} = V_{REF} \cdot \left(1 + \frac{R_2}{R_1}\right)$ is provided in the datasheet to calculate resistor values R1 and R2. V_{REF} has a value of $1{,}23\ V$, V_{out} is the desires output of $5\ V$ and R_1 should be approximately $1\ k\Omega$. R_2 is calculated to be $3065\ \Omega$. A more conventional resistance of $3\ k\Omega$ provides an output voltage of $4{,}92\ V$.
The LM2596-5.0 fits in the design by shorting R_2 or replacing it with a low resistance resistor as displayed in Figure 23. Resistor R_1 is left as an open connection. The LM2596-5.0's output is a proper five volts.
(Texas Instruments 2016)

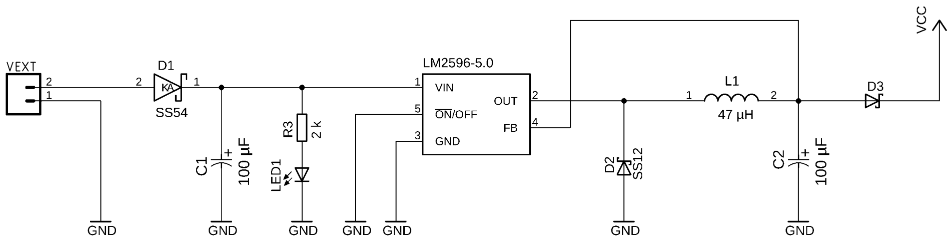

Figure 23. LM2596-5.0 scheme

4.5 Power supply switch

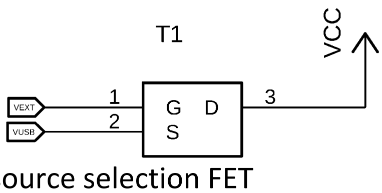

Figure 24. Power source selection FET

Power sources are automatically selected through a p-channel field effect transistor (FET). The function is as following: the transistor is conducting from source to drain when no voltage is applied on the gate. V_{EXT} is controlling the gate and connected to the drain. V_{USB} is applied on the source. When no external voltage is applied over the terminal, the FET conducts V_{USB} to the drain. The transistor switches off the USB power

when the regulator outputs a voltage higher than the $3\ V$ gate to source threshold voltage.
(Vishay 2008)

5 PCB

A custom PCB is designed with EAGLE to accommodate the components. Traces are places following the schematics. The size and hole placement are in accordance to the raspberry pi formfactor, credit card sized.

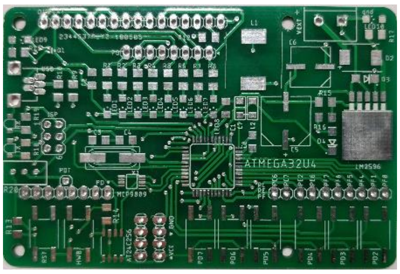

Figure 25. PCB Front

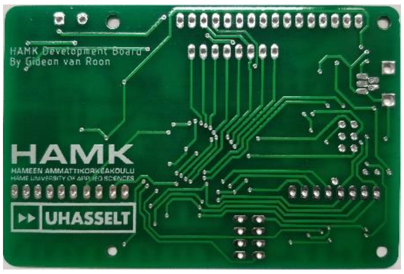

Figure 26. PCB backside

The $1{,}6\ mm$ thick PCB is designed with two layers: component side and bottom side. Both sides are displayed in Figure 25 and Figure 26.
Traces are $20\ mil$ in width except for the trace from $VEXT$ to the regulator input. This trace can handle more power with a width of $40\ mil$. Copper thickness is $35\ \mu m$. The small traces can withstand up to $3\ A$ if allowed to heat up to $50°C$. The $40\ mil$ trace can handle up to $5\ A$. The maximum current is calculated according to the IPC-2221 standard. $I = k\Delta T^{0{,}44} \cdot A^{0{,}725}$ is used with the current I, constant $k = 0{,}024$ for internal layers, ΔT the temperature difference and A the cross-section area.
(IPC 1998)
(Bittile Electronics Inc n.d.)
Via drill holes are $0{,}35\ mm$ in diameter and the pads have a diameter of $0{,}75\ mm$.
SMD component are applied where possible. The only non-smd parts are the headers, the terminal and the USB port.
Soldering is performed by hand. The order of soldering is no concern but easiest to begin with small components. The smallest components are picked in the imperial 1206 format only for the purpose of manual soldering.
Design rule checks are performed in compliance with the technology maintained by the company responsible for manufacturing.
(JIALICHUANG Electronic Technology Development Co n.d.)

6 THERMAL

Thermal tests are conducted to determine the power capabilities. A FLIR ONE pro thermal camera is used to gather thermal data. Two sensors collect data; a camera for an overall image and an infrared sensor to gain temperature readings. The two data sets are then combined. A parallax effect occurs due to the two sensors and the short measurement distance. Figure 27 displays a thermal image. Some components are made visible in the Thermal layer. The components are shifted up relative to the visual image. Accuracy is a typical $\pm 3°C$ and a maximum $\pm 5°C$.
(FLIR® Systems n.d.)

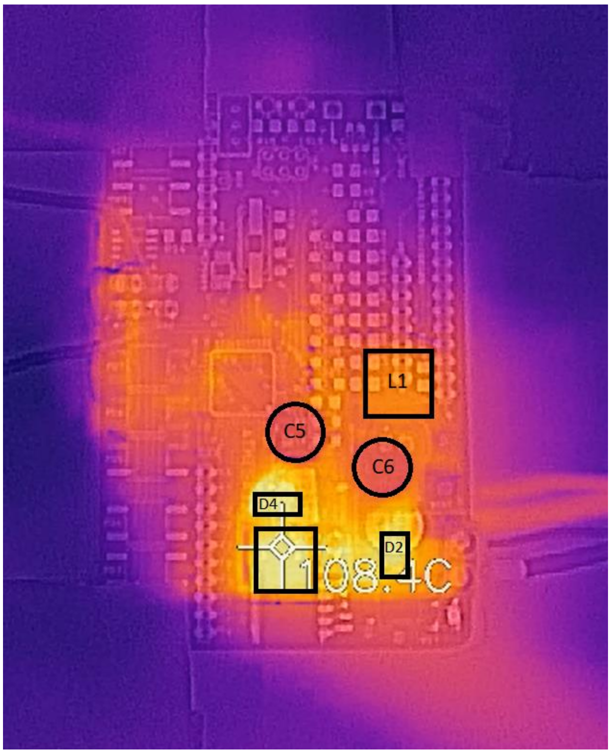

Figure 27. Thermal image of the board under load

The white cross indicates the reading spot. All readings are performed on the LM2596-ADJ regulator.
Table 1 and Table 2 show readings from the power supply in function of a certain applied voltage on the input. Readings contain the supplied amps, power and settle temperature. The settle temperature is the regulators temperature after fifteen minutes under constant load. Table 1 displays under a 3 Ω load and Table 2 displays a 22 Ω load.

Vin	Amps [A]	Power [W]	Settle Temp
6V	1,3	7,8	121°C
12V	0,81	9,72	119°C
24V	0,28	6,72	92°C

Table 1. Amps, Power drain and Settle temperature for given input voltage and 3 Ω resistance.

The temperature increases when more waste power is dissipated. Table 1 confirms the behaviour plotted in Figure 15; A higher input voltage results in a higher efficiency thus converting less power into heat. The maximum junction temperature is $150°C$ and maximum operating temperature $125°C$.
(Texas Instruments 2016)
During the fifteen-minute test, the temperature reached $100°C$ after two minutes and twenty-six seconds. $120°C$ was reached after eight minutes and ten seconds. The regulator operated the remaining seven minutes around the settling temperature. A cooling solution should be applied when higher loads are demanded.

Vin	Amps [A]	Power [W]	Settle Temp
6V	0,24	1,44	49°C
12V	0,13	1,56	55°C
24V	0,08	1,92	49°C

Table 2. Amps, Power drain and Settle temperature for given input voltage and 22 Ω resistance.

Table 2 contains reading with a $22\ \Omega$ resistance as load. The power drawn is significantly lower resulting in lower temperatures.

	Vin	Amps [A]	Power [W]	Settle Temp	T_125°C [s]
uncooled:	3,8V	3,02	11,476	-	41
cooled:	3,8V	3,02	11,476	108,8°C	-

Table 3. Readings at a cooled and uncooled, shorted circuit

It is recommended to prevent short circuits at any moment. When a short circuit occurs due to a mistake, the board circuit can withstand the load for a small amount of time. Measurements displayed in Table 3 show that a shorted regulator without cooling reaches $125°C$ after forty-one seconds. The maximum operating temperature was not met, after fifteen minutes, with a fan blowing over the board as cooling solution.

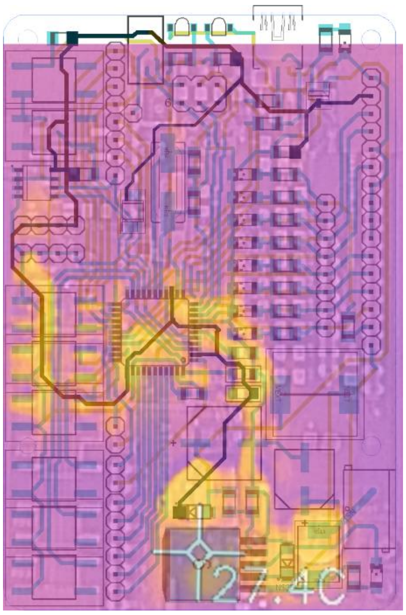

Figure 28. Heated VCC track

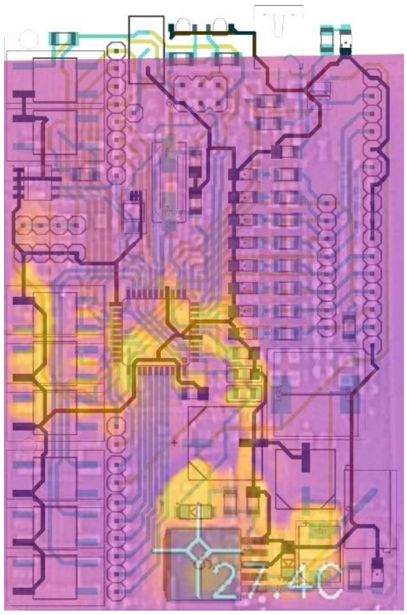

Figure 29. Heated GND track

PCB traces heat up at power transmission. Figure 28 and Figure 29 show respectively the VCC and GND tracks with a thermal image overlay. The traces are noticeable on the thermal image but do not overheat.

7 COST ANALYSIS

Prices are calculated in price per unit according to pricing on Farnell. Large volume orders drop the unit price because volume discounts apply on most components. Another advantage is that less components only available in bulk packages remain unused.
Unit prices are calculated in quantities of 1, 10 and 100 boards.

7.1 MCU

Internal USB controllers avoid the additional cost of a programmer. Comparable PIC processors might be cheaper, however including a programmer makes the overall combination rather expensive.
Table 4 contains combinations with three popular microcontrollers and the ATmega8U2 as USB controller. The ATmega8U2 is the cheapest Atmel controller capable of maintaining a USB connection. The main difference between the ATmega32U4 and ATmega16U4 is the amount of programmable flash memory, 32U4 has 32 kbyte and the 16U4 16 kbyte. The price difference is minor compared to the amount is extra program memory.
ATmega328-PU is a controller with comparable features to the ATmega32U4 but lacking the USB controller. The total cost in combination with the ATmega8U2 as USB controller is also minor. The PCB area would at least double and the 328-PU is available in dual in-line package (DIP) format only.
(Microchip Technology 2016)

Microcontroller	USB Controller	€ MCU	€ USB	€ Total
ATmega32U4	Integrated	3,61 €		3,61 €
	ATmega8U2	3,61 €	1,94 €	5,55 €
ATmega16U4	Integrated	3,52 €		3,52 €
ATmega328-PU	ATmega8U2	1,74 €	1,94 €	3,68 €

Table 4: Microcontroller comparison table

7.2 PSU

The power supply be composed out of two regulators; The LM2596-5.0 and LM2596-ADJ. Price differences are significant. LM2596-ADJ’s are only €1,83 while the LM2596-5.0’s start from €4,40. An additional resistor costing €0,01 must be combined with the LM2596-ADJ. Ordering in quantities of 10 and 100 result in a cheaper price per unit.

	Price/unit	Price/10	Price/100
LM2596-5.0:	4,40 €	3,55 €	2,91 €
LM2596-ADJ:	1,83 €	1,83 €	1,26 €

Table 5. LM2569 prices per unit, 10 units and 100 units

Table 5 lists the volume prices for both components. The price per unit in quantities of 100 is about $66\% - 69\%$ compared to single unit prices.

	Price/unit	Price/10	Price/100
Total:	17,49 €	59,90 €	508,40 €
PPU:	17,49 €	5,99 €	5,08 €

Table 6. LM2596-5.0 total power supply cost and price per unit

	Price/unit	Price/10	Price/100
Total:	14,99 €	42,77 €	343,77 €
PPU:	14,99 €	4,28 €	3,44 €

Table 7. LM2596-ADJ total power supply cost and price per unit

Total power supply cost drops remarkable due to two main reasons. First, almost no components will be left unused. Second, the previous discussed quantity discount. The result is a single unit price that is almost three time higher than ten units.
The power supply is responsible for about 20% of the total price. Lowering in price is less than the average of all components combined, causing the share to rise when increasing the unit amount.

	Price/unit	Price/10	Price/100
%Total Price	17%	20%	22%

Table 8. Power supply share per unit

Costs can be further lowered with better knowledge about the load that will be applied in the courses.
(Premier Farnell n.d.)

7.3 **PCB**

PCB's are produced by JLCPCB. The PCB contains three costs. The most expensive one: the shipping. The second most expensive is the lead-free HASL and the least expensive the boards themselves.

The percentage of the total cost is huge compared to any other part when producing one unit. The minimum order amount is 10 pieces, causing the price to be equal for a single unit and for ten units.

	Price/unit	Price/10	Price/100
%Total Price:	29%	12%	5%

Table 9. PCB Share per unit

The main expense for ten units is due to shipping.

	Price/unit	Price/10	Price/100
PCB:	1,62 €	1,62 €	36,54 €
Lead-free HASL:	4,86 €	4,86 €	4,05 €
Shipping:	19,31 €	19,31 €	31,99 €

Table 10. PCB price details

(JIALICHUANG Electronic Technology Development Co n.d.)

7.4 **Total cost**

The total cost follows the general rule: larger orders result in a lower cost per unit.

	Price/1 board	Price/10 boards	Price/100 boards
Total:	87,62 €	214,59 €	1598,15 €
PPU:	87,62 €	21,46 €	15,98 €

Table 11. Total cost for one, ten and one hundred units

A single unit cost as much as € 87,62. When ten units are ordered, the prices drop below one third of the single unit price: € 21,46. The drop is less significant when ordering one hundred units. The price per unit is then only €15,87.

8 SOFTWARE

Programming and uploading sketches are done entirely in Atmel Studio. Executing programs written in the Arduino dialect can also be written and uploaded through Atmel Studio if desired. It must be noted that not all GPIO ports are accessible in the Arduino framework. For example, PB0 and PD5 are used to drive the TX and RX LEDs on Arduino boards.
AVR c language is preferred over the Arduino c language. Arduino made programming very easy and quick for prototyping. Full control over the processor can only be gained in AVR c as Arduino prepares large parts of the setup and structure in the background.

8.1 Programming with Atmel Studio

Coding, compiling and uploading are done within Atmel Studio.
When creating a new project, several options must be picked. Programming can be done in assembler, but many prefer C/C++. The project type is most of the time a “GCC C Executable Project”. Next, the device selection window presents itself. This window questions which microcontroller the code has to be compiled for. Naturally this is the ATmega32U4.
A project is created containing a main function with an empty while loop in it. The project can be edited to obtain the whished functionality.
To upload the program over DFU, the bord must enter the bootloader section. Connect the board over USB to the computer. Press both RST and HWB. Release RST first until the device is connected. To check if the board is available, check under Tools > Device Programming, in the tool dropdown menu if the Device Firmware uploader is available.
If the previous steps are completed, the programs can be compiled and uploaded in four different methods. First, press the hollow arrow button in the top ribbon. Second, select Debug > Start Without Debugging. The hotkey ctrl+alt+F5 can be used to start. Last, to upload any .elf or .hex file, open the Device Programming window and go to “Memory”. In memory, a section called “Flash”, a path selector can be used to navigate to the .elf or .hex file. The file is uploaded by “program”.
(Microchip Technology 2016)

8.2 Buttons and LEDs

The following code mimics the buttons on the LEDs.

```
#include <avr/io.h>

int main(void) {

   DDRB = 0xff;   //PORTB output
```

```
    PORTB = 0xff;  //PORTB '1'
    DDRD = 0x00;   //PORTD input
    PORTD = 0xff;  //PORTD internal pull up

    while(1)
    {
                PORTB = ~PIND;
    }
}
```

The lines can be explained as followed:

`#include <avr/io.h>` imports the io library, necessary for reading and driving the inputs and outputs.

Atmel Studio maintains the following structure:

```
int main(void) {

    while(1)
    {

    }
}
```

The main is the first object in the program memory and is executed on boot when HWB is not pressed. In the main is a while located with condition "1". The while will always be "1" or true and remains looping its content.

Initialising happens in between the main and while function. Code placed in this section is only executed once, on boot.

The microcontroller must know what ports are used as inputs and outputs to read the buttons and drive the LEDs.

```
  DDRB = 0xff;   //PORTB output
  PORTB = 0xff;  //PORTB '1'
  DDRD = 0x00;   //PORTD input
  PORTD = 0xff;  //PORTD internal pull up
```

DDRx, or Data Direction of Register x indicates the data direction. The hexadecimal notation $0xff$ holds a single byte with each bit representing a high signal. All bits are low in $0x00$. High is used for an outgoing and low for an incoming data direction.

PORTx enables the pull-up resistors. The port is in a tri-state when not applied and thus can be any value. A high signal is received with the pull-up resistor applied.

The next line assigns the inverted reading from buttons to the LEDs.

```
  PORTB = ~PIND;
```

Inverting is applied because the buttons are active high due to the internal pull-up resistor.

(Microchip Technology 2007)

8.3 **LCD**

The LCD screen is driven by the Hitachi hd44780 or compatible controller. A library by SA development is used to communicate with the controller. In the library are three files: hd44780.h, hd44780.c and hd44780_settings. The pins used on the board header must be matched in the settings file. The part of the settings file that must be changed is located in the appendix.
After the ports are setup correctly, the backlighting and contrast must be controlled. Backlight and contrast are driven on pins that are both PWM compatible. A duty cycle of 25% is recommended for the contrast.
A line is printed in three easy commands:

```
lcd_init();
lcd_clrscr();
lcd_puts("Hello HAMK");
```

First, the LCD settings are initialized. Then the screen is cleared and finally the text in between the quotes is written to the screen.
(Microchip Technology 2011)
(Nongnu 2016)
(Peter Fluery 2006)

8.4 **I²C**

Communication contains two cases: the master sends data to the slave or the master requests data from the slave.
To send data to the slave; First the start condition is met (pulling the data line low while the clock is high, Figure 30), and the master sends the unique slave address. The slave notices the address and listens for the data that is send next. The stop condition (releasing the data line while the clock is high, Figure 30 is instantiated after the transfer is done.

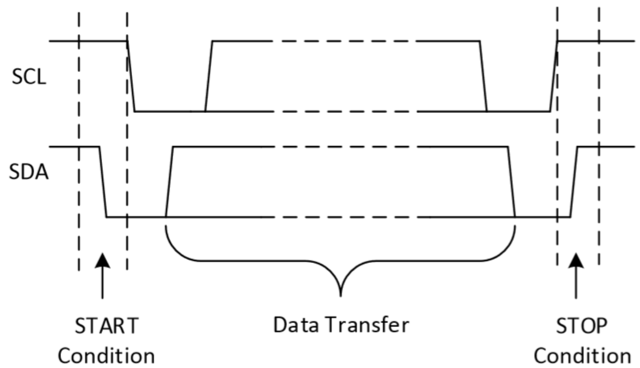

Figure 30: I²C Start condition, data transfer and stop condition (Texas Instruments 2015)

Requesting data from a device uses the same Start and stop conditions. The intermediate steps differ from writing to a chip. After the start condition, the requested register is send. Next, the register is sent back to the master. The stop condition is met at last.

(NXP Semiconductors 2014)
(Texas Instruments 2015)
A library is used to set up and maintain the communication according to the protocol. The library used is created by Peter Fleury and works as following; First the device addresses are needed including the read and write bits. These are represented in hexadecimal notation in Table 12.

	write	read
MCP9808	30	31
AC24C	A0	A1

Table 12: Hexadecimal write and read addresses

```
i2c_init();
```

sets up the communication, for example the transmission frequency. Next, a specific device is contacted, and the start condition is met to read or write. For example, to write byte AA to register 05 and terminate the connection two writes are used:

```
i2c_start_wait(DeviceAdressWrite);
i2c_write(0x05);
i2c_write(0xAA);
i2c_stop();
```

First, the device is acknowledged that a connection is set up. Followed by two writes; One for the address in question and second the data that is written to it.
To read data from register 05:

```
i2c_start_wait(DeviceAdressWrite);
i2c_write(0x05);
i2c_rep_start(DeviceAdressRead);
retrieved = i2c_readNak();
i2c_stop();
```

The register that is written is sent to the device, then the master waits to receive the data. This data is then stored in to the retrieved parameter.
(Microchip Technology n.d.)
(Peter Fluery 2006)

CONCLUSION

A development board is realised. The microcontroller driving the board is an ATmega32U4. Peripherals consist out: six buttons, eight LEDs, an IR emitter, an IR receiver, an I^2C MCP9808 temperature sensor and an I^2C AT24C256C memory chip. All ports are available over female headers including a dedicated LCD display header. A potentiometer provides a variable voltage output.

The board can be powered over USB or via an external applied voltage up to 40V.

All components are collected on a credit card sized PCB.

Programming and uploading software can be exclusively done within Atmel studio.

Prices range from € 87,62 for a single unit down to € 15,87 per unit for an order of 100 units.

REFERENCES

Arduino (2015). display() and noDisplay() methods. Retrieved 25 March 2018 from https://www.arduino.cc/en/Tutorial/LiquidCrystalDisplay

Bittile Electronics Inc (n.d.). PCB Trace Width Calculator. Retrieved 01 April 2018 from https://www.7pcb.com/trace-width-calculator.php

Engbedded (2014). Engbedded Atmel AVR® Fuse Calculator. Retrieved 04 May 2018 From http://www.engbedded.com/fusecalc/

FLIR® Systems (n.d.). FLIR ONE PRO. Retrieved 23 May 2018 from http://www.flir.eu/flirone/pro/

IPC (1998). Generic Standard on Printed Board Design. Retrieved 03 April 2018 from http://www.ipc.org/TOC/IPC-2221A.pdf

Jerry Seams (2007). Terminating transmission lines. Retrieved 01 June 2018 from https://www.electronicproducts.com/Passive_Components/Resistors_and_Potentiometers/Terminating_transmission_lines.aspx

JIALICHUANG Electronic Technology Development Co (n.d.). Capabilities. Retreived 15 April 2018 from https://jlcpcb.com/capabilities/Capabilities

Larry Davis (n.d.). Trace termination. Retrieved 01 June 2018 from http://www.interfacebus.com/Design_Termination.html

Microchip Technology (n.d.). <avr/fuse.h> Fuse Support. Retrieved 02 May 2018 from https://www.microchip.com/webdoc/AVRLibcReferenceManual/group__avr__fuse.html

Microchip Technology (2011). ±0.5°C Maximum Accuracy Digital Temperature Sensor. Retrieved 27 February 2018 from http://ww1.microchip.com/downloads/en/DeviceDoc/25095A.pdf

Microchip Technology (n.d.). Microchip® Advanced Software Framework. Retrieved 16 may 2018 from http://asf.atmel.com/docs/latest/common.components.sensors.mcp980x.twi_temp_sensor_example.sam3u_ek/html/files.html

Microchip Technology (2016). ATmega16U4/ATmega32U4. Retrieved 05 February 2018 from http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-7766-8-bit-AVR-ATmega16U4-32U4_Datasheet.pdf

Microchip Technology (n.d.). AVR Assembler Instructions. Retrieved 05 June 2018 from https://www.microchip.com/webdoc/avrassembler/avrassembler.wb_instruction_list.html

Microchip Technology (2011). HD44780 Library. Retrieved 21 May 2018 from https://community.atmel.com/projects/hd44780-library

Microchip Technology (2015). I2C-Compatible (2-Wire) Serial EEPROM. Retrieves 29 February from http://ww1.microchip.com/downloads/en/DeviceDoc/Atmel-8568-SEEPROM-AT24C256C-Datasheet.pdf

Microchip Technology (2009). LCD Tutorial. Retrieved 21 May 2018 from https://www.avrfreaks.net/forum/tut-lcd-tutorials-updated

Microchip Technology (n.d.). lcd.c. Retrieved 20 May 2018 from https://www.microchip.com/webdoc/AVRLibcReferenceManual/group__stdiodemo_1stdiodemo_lcd_c.html

Microchip Technology (2018). mcp980x.c File Reference. Retrieved 18 May 2018 from http://asf.atmel.com/docs/latest/common.components.sensors.mcp980x.twi_temp_sensor_example.sam3u_ek/html/mcp980x_8c.html

Microchip Technology (2007). Push button to light led. Retrieved 15 May 2018 from https://www.avrfreaks.net/forum/push-button-light-led

Microchip Technology (2008). USB DFU Bootloader Datasheet. Retrieved 09 February 2018 from http://ww1.microchip.com/downloads/en/DeviceDoc/doc7618.pdf

Multicomp (2011). Schottky Barrier Rectifier. Retrieved 10 March 2018 from http://www.farnell.com/datasheets/1441694.pdf

Multicomp (2011). Schottky Barrier Rectifier. Retrieved 10 March 2018 from http://www.farnell.com/datasheets/1441695.pdf

Nongnu (2016). <avr/fuse.h> Fuse Support. Retrieved 04 May 2018 from https://www.nongnu.org/avr-libc/user-manual/group__avr__fuse.html

Nongnu (2016). Using the standard IO facilities. Retrieved 15 May 2018 from https://www.nongnu.org/avr-libc/user-manual/group__stdiodemo.html

NXP Semiconductors (2014). I²C-bus specification and user manual. Retrieved 01 June 2018 from https://www.nxp.com/docs/en/user-guide/UM10204.pdf

Objective Development Software GmbH (2012). V-USB. Retrieved 09 February 2018 from https://www.obdev.at/products/vusb/index.html

Patterson, D. & Hennessy, J. (2014). *Computer organisation and design.* New York: Elsevier.

Peter Fluery (2006). AVR-Software. Retrieved 16 May from http://homepage.hispeed.ch/peterfleury/avr-software.html

Portland State University (n.d.). Transmission Lines, Reflections, and Termination. Retrieved 01 June 2018 from http://web.cecs.pdx.edu/~greenwd/xmsnLine_notes.pdf

Premier Farnell (n.d.). Retrieved 06 February 2018 from http://www.farnell.com/

Texas Instruments (2016). LM2596 SIMPLE SWITCHER® Power Converter 150-kHz 3-A Step-Down Voltage Regulator. Retrieved 10 March 2018 from http://www.ti.com/lit/ds/symlink/lm2596.pdf

Texas Instruments (2015). Understanding the I²C Bus. Retrieved 01 June 2018 from http://www.ti.com/lit/an/slva704/slva704.pdf

Texas Instruments (n.d.). WEBENCH® Design Center. Retrieved 14 May 2018 from https://webench.ti.com

USB-IF (n.d.). USB 2.0 Frequently Asked Questions. Retrieved 01 June 2018 from http://www.usb.org/developers/usbfaq#sig6

Vishay (2018). 3/8" Square Multi-Turn Cermet Trimmer. Retrieved 05 April 2018 from https://www.vishay.com/docs/51026/t93.pdf

Vishay (2017). High Speed Infrared Emitting Diode, 940 nm, Surface Emitter Technology. Retrieved 28 February 2018 from https://www.vishay.com/docs/84253/vsmy2943sl.pdf

Vishay (2008). P-Channel 30-V (D-S) MOSFET. Retrieved 23 April 2018 from https://www.vishay.com/docs/72699/72699.pdf

Vishay (2013). Silicon NPN Phototransistor. Retrieved 28 February 2018 from https://www.vishay.com/docs/84166/vemt2023slx01.pdf

AVR Comparison Chart

8-bit AVR® Microcontrollers Peripheral Integration

Quick Reference Guide

Peripheral Function Focus

Product Family	Pin Count	Program Flash Memory (KB)	SRAM (KB)	Intelligent Analog						
				ADC (# of bits)	ADC (# of channels)	Comparators	ADC Gain Stage	DAC (# of bits)	Temperature Sensor	Internal Voltage Reference
ATtiny4/5/9/10	6	0.5–1	0.032	10[3]	4[3]	✓				
ATtiny102/104	8/14	1	0.032	10	5/8	✓				✓
ATtiny13A	8–20	1	0.064	10	4	✓				
ATtiny20/40	12–20	2/4	0.128/0.256	10	8/12	✓			✓	
ATtiny24A/44A/84A	14–20	2–8	Up to 0.512	10	8	✓	✓		✓	✓
ATtiny25(V)/45(V)/85(V)	8–20	2–8	Up to 0.512	10	4	✓	✓		✓	✓
ATtiny48/88	28–32	4/8	Up to 0.512	10	8	✓			✓	✓
ATtiny87/167	20–32	8/16	0.512	10	11	✓			✓	✓
ATtiny261A/461A/861A	20–32	2–8	Up to 0.512	10	11	✓	✓		✓	✓
ATtiny20x/40x/80x/160x	8–24	2–16	Up to 1	10	12	✓			✓	✓
ATtiny21x/41x/81x/161x/321x	8–24	2–32	Up to 2	10	12	✓		8	✓	✓
ATtiny441/841	14–20	4/8	Up to 0.512	10	12	✓	✓		✓	
ATtiny1634	20	16	1	10	12	✓			✓	✓
ATtiny2313A	20	2	0.128	–	–	✓				✓
ATmega8A/16A/32A	28–44	8–32	1–2	10	8	✓				
ATmega8U2/16U2/32U2	32	8–32	0.5–1	–	–	✓			✓	✓
ATmega16U4/32U4	32	16/32	1/2	10	12	✓			✓	✓
ATmega48PB/88PB/168PB/328PB	32	4–32	0.5–2	10	8	✓			✓	✓
ATmega320x/480x	28–48	32–48	Up to 6	10	16	✓			✓	✓
ATmega64A/128A	64	64–128	4	10	8	✓	✓			✓
ATmega164PA/324PA/644PA/1284P	44	16–128	1–16	10	8	✓	✓			✓
ATmega165PA/325PA/645P	44	16–64	1–4	10	8	✓				✓
ATmega169PA/329PA/649P	64	16–64	1–4	10	8	✓				✓
ATmega324PB	44	32	2	10	8	✓				✓
ATmega640/1280/2560/1281/2561	64–100	64–256	8	10	8/16	✓	✓			✓
ATmega3290PA/6490P	100	32–64	2–4	10	8	✓	✓			✓
ATmega3250PA/6450P	100	32–64	2–4	10	8	✓	✓			✓
ATxmega A1U Family	100	64–128	4–8	12	16	✓	✓	12	✓	✓
ATxmega A3U Family	64	64–256	4–16	12	16	✓	✓	12	✓	✓
ATxmega A4U Family	44–49	16–128	2–8	12	12	✓	✓	12	✓	✓
ATxmega B1/B3 Family	64–100	64–128	4–8	12	8	✓	✓		✓	✓
ATxmega C3/D3 Family	64	32–384	4–32	12	16	✓	✓		✓	✓
ATxmega C4/D4 Family	44–49	16–128	2–8	12	12	✓	✓		✓	✓
ATxmega E5 Family	32	8–32	1–4	12	16	✓	✓	12	✓	✓

Product Family	Waveform Control				Timing and Measurements				Logic, Crypto and Math			Safety and Monitoring			
	8-bit PWM	16-bit PWM	Quadrature Decoder	Waveform Extension	Real-Time Counter	8-bit Timer/Counters	12-bit Timer Counter	16-bit Timer/Counter	CCL	MULT	Crypto (AES/DES)	CRC	POR	BOD	WDT
ATtiny4/5/9/10		2						1					✓		✓
ATtiny102/104		2						2					✓		✓
ATtiny13A		2											✓	✓	✓
ATtiny20/40	2	2				1		1					✓	✓	✓
ATtiny24A/44A/84A	2	2				1		1					✓	✓	✓
ATtiny25(V)/45(V)/85(V)	4					2							✓	✓	✓
ATtiny48/88	1	1				1		1					✓	✓	✓
ATtiny87/167	1	2				1		1					✓	✓	✓
ATtiny261A/461A/861A						1		1					✓	✓	✓
ATtiny20x/40x/80x/160x		2			✓			1	✓	✓		✓	✓	✓	✓
ATtiny21x/41x/81x/161x/321x		2			✓		1	1	✓	✓		✓	✓	✓	✓
ATtiny441/841	1	2				1		2					✓	✓	✓
ATtiny1634	2	2				1		1					✓	✓	✓
ATtiny2313A	2	2				1		1					✓	✓	✓
ATmega8A/16A/32A	2	1			✓	2		1		✓			✓	✓	✓
ATmega8U2/16U2/32U2	4	6			✓	2		3		✓			✓	✓	✓
ATmega16U4/32U4	5					1		1		✓			✓	✓	✓
ATmega48PB/88PB/168PB/328PB	4	2/6[6]			✓	2		1/3[6]		✓			✓	✓	✓
ATmega320x/480x	4	3			1			5	✓	✓		✓	✓	✓	✓
ATmega64A/128A	2	6				2		2		✓			✓	✓	✓
ATmega164PA/324PA/644PA/1284P	4	2/2/4			✓	2		1/1/2		✓			✓	✓	✓
ATmega165PA/325PA/645P	4	6			✓	2		3		✓			✓	✓	✓
ATmega169PA/329PA/649P	2	2			✓	2		1		✓			✓	✓	✓
ATmega324PB	2	2			✓	2		1		✓			✓	✓	✓
ATmega640/1280/2560/1281/2561	4	6/12			✓	2		4		✓			✓	✓	✓
ATmega3290PA/6490P	2	2			✓	2		1		✓			✓	✓	✓
ATmega3250PA/6450P	2	2			✓	2		1		✓			✓	✓	✓
ATxmega A1U Family		8	✓	✓	✓			8		✓	✓	✓	✓	✓	✓
ATxmega A3U Family		7	✓	✓	✓			7		✓	✓	✓	✓	✓	✓
ATxmega A4U Family		5	✓	✓	✓			5		✓	✓	✓	✓	✓	✓
ATxmega B1/B3 Family		2/3	✓	✓	✓			2/3		✓	✓	✓	✓	✓	✓
ATxmega C3/D3 Family		5	✓	✓	✓			5		✓		✓	✓	✓	✓
ATxmega C4/D4 Family		4	✓	✓	✓			4		✓		✓	✓	✓	✓
ATxmega E5 Family		3	✓	✓	✓			3	✓	✓		✓	✓	✓	✓

Product Family	Communications						User Interface			System Flexibility					
	UART	USART	USB	I²C	SPI	IRCOM	QTouch® Technology	QTouch Technology with PTC[2]	LCD	External Bus Interface	DMA Channels	Event System	SleepWalking	Sleep Modes	picoPower® Technology
ATtiny4/5/9/10							✓							4	
ATtiny102/104		1												4	
ATtiny13A							✓							3	✓
ATtiny20/40				1	1		✓							4	
ATtiny24A/44A/84A				1	1		✓							4	✓
ATtiny25(V)/45(V)/85(V)				1	1		✓							3	
ATtiny48/88				1	1									3	✓
ATtiny87/167	1[1]			1	2									4	
ATtiny261A/461A/861A				1	1		✓							4	✓
ATtiny20x/40x/80x/160x		1[1]		1	1							✓	✓	3	✓
ATtiny21x/41x/81x/161x/321x		1[1]		1	1			✓[4]				✓	✓	3	✓
ATtiny441/841		2		1	1									4	✓
ATtiny1634		2		1				✓						4	✓
ATtiny2313A		1		1	2									3	✓
ATmega8A/16A/32A		1		1	1		✓							5	
ATmega8U2/16U2/32U2		2	✓	2	2			✓						6	
ATmega16U4/32U4		1	✓		1									6	
ATmega48PB/88PB/168PB/328PB		1/2[6]		1/2[6]	1/2[6]		✓	1[6]						6	
ATmega320x/480x	4			1	1								✓	3	✓
ATmega64A/128A		2		1	1		✓							6	
ATmega164PA/324PA/644PA/1284P		2		1	1		✓							6	✓
ATmega165PA/325PA/645P		3		2	2			✓						6	✓
ATmega169PA/329PA/649P		1		1	1		✓		✓					5	
ATmega324PB		1		1	1		✓	✓						5	
ATmega640/1280/2560/1281/2561		2/4		1	1		✓			✓[5]				6	
ATmega3290PA/6490P		1		1	1		✓		✓					5	
ATmega3250PA/6450P		1		1	1		✓							5	
ATxmega A1U Family		8	✓	4	4	✓	✓			✓	4	✓		5	✓
ATxmega A3U Family		7	✓	2	3	✓	✓				4	✓		5	✓
ATxmega A4U Family		5	✓	2	2	✓	✓				4	✓		5	✓
ATxmega B1/B3 Family		1/2	✓	1	1	✓	✓		✓		2	✓		5	✓
ATxmega C3/D3 Family		3	✓[7]	2	2	✓	✓					✓		5	✓
ATxmega C4/D4 Family		2	✓[7]	2	2	✓	✓					✓		5	✓
ATxmega E5 Family		2		1	1	✓	✓				4	✓		5	✓

1: LIN port also 2: Peripheral Touch Controller 3: Only on the ATtiny5/10 4: Not on the ATtiny212/214/412/414/416 5: Only on the ATmega1281/2561 6: Only on the ATmega328PB 7: Only on the C3 and C4

www.microchip.com/8bit

Component values

Resistors		Capacitors		Inductors		Diodes		Headers	
R1 - R8	220	C1	1µ	L1	47µ	D2	SS54	Display	16 pin
R9	22	C2	10µ	L2	10µ	D3	SS12	VCC	4 pin
R10	22	C3	22p			D4	SS12	GND	4 pin
R11	220	C4	22p					PB	8 pin
R12	220	C5	100µ					PC	2 pin
R13	10k	C6	100µ					PD	8 pin
R14	10k	C7	100n					PE	2 pin
R15	1k or open	C8	100n					PF	6 pin
R16	3k or 0								
R17	2K								
R18	220								
R19	10k								
R20	10k pot								
R21 -R22	4k7								

Note: Component indicators might not apply to schematics used in this paper. The numbers comply with the numbers on the PCB.

Appendix 3

Bill of materials: Board

	Component		Price/unit	Price/10	Price/100	units	Price/1 board	Price/10 boards	Price/100 boards
Main:	MCU:	Atmega32U4	3,61	3,61	2,89	1	3,61 €	36,10 €	289,00 €
	Resistors:	22		0,0106	0,0063	3	0,11 €	0,32 €	1,89 €
		220		0,0111	0,0067	10	0,11 €	0,67 €	6,70 €
		2k		0,0105	0,0063	1	0,11 €	0,11 €	0,63 €
		4k7		0,0106	0,0078	2	0,11 €	0,21 €	1,56 €
		10k		0,0105	0,0063	2	0,11 €	0,21 €	1,26 €
	Caps:	22p		0,096	0,091	2	0,96 €	1,92 €	18,20 €
		1µ		0,1406	0,1034	3	1,41 €	4,22 €	31,02 €
		10µ		0,009	0,008	1	0,09 €	0,09 €	0,80 €
		0,1µ		0,0822	0,037	1	0,82 €	0,82 €	3,70 €
	Inductor	10µ		0,219	0,138	1	2,19 €	2,19 €	13,80 €
	Crystal	16MHz	0,47	0,47	0,26	1	0,47 €	4,70 €	26,00 €
	USB:	Mini	1,1	1,1	1	1	1,10 €	11,00 €	100,00 €
	LED:			0,261	0,141	10	2,61 €	14,10 €	141,00 €
	IR:	Transmitter		0,27	0,259	1	2,70 €	2,70 €	25,90 €
		Receiver		0,58	0,384	1	5,80 €	5,80 €	38,40 €
	Buttons:			0,183	0,183	6	1,83 €	10,98 €	109,80 €
	POT:	10k	1,41	1,2	0,927	1	1,41 €	12,00 €	92,70 €
	T_Sensor:	MCP9808	1,12	1,03	0,739	1	1,12 €	10,30 €	73,90 €
	Memory:	AT24C256C	0,446	0,446	0,376	1	0,45 €	4,46 €	37,60 €
	Headers:	2x3 Male		0,239	0,224	1	2,39 €	2,39 €	22,40 €
		1x2 Female			0,0658	1	3,29 €	3,29 €	6,58 €

		1x4 Female		0,103	0,076	2	1,55 €	3,09 €	17,58 €
		1x6 Female			0,123	2	1,85 €	3,69 €	24,60 €
		1x8 Female			0,173	1	2,60 €	2,60 €	17,30 €
		1x16 Female		0,808	0,679	1	8,08 €	8,08 €	67,90 €
Power Sup:	Elco:	100µ	0,701	0,389	0,28	2	0,70 €	3,89 €	56,00 €
	Diode:	SS12		0,0708	0,051	2	0,71 €	1,42 €	10,20 €
		SS54		0,241	0,182	1	2,41 €	2,41 €	18,20 €
	Inductor:	47µH	0,825	0,59	0,487	1	0,83 €	5,90 €	48,70 €
	Resistor:	2k		0,028	0,022	1	0,28 €	0,28 €	2,20 €
	Transistor:	SI2307		0,467	0,309	1	2,34 €	4,67 €	30,90 €
	Terminals:	1x2		0,567	0,502	1	5,67 €	5,67 €	50,20 €
	Regulator:	LM2596-ADJ	1,83	1,83	1,26	1	1,83 €	18,30 €	126,00 €
	Resistors:	1k		0,0112	0,0067	1	0,11 €	0,11 €	0,67 €
		3k		0,0117	0,007	1	0,12 €	0,12 €	0,70 €
PCB:	PCB:			1,62	36,54	1	1,62 €	1,62 €	36,54 €
	Lead-free HASL:			4,86	4,05	1	4,86 €	4,86 €	4,05 €
	Shipping:			19,31	31,99	1	19,31 €	19,31 €	31,99 €
							+	+	+
						Total:	87,62 €	214,59 €	1 586,57 €
						PPU:	87,62 €	21,46 €	15,87 €
							+	+	+
						Total:	77,24 €	195,01 €	1 440,47 €
						PPU:	77,24 €	19,50 €	14,40 €

Bill of materials: PSU

	Component		Price/unit	Price/10	Price/100	units	Price/1 board	Price/10 boards	Price/100 boards
Power Sup:	Elco:	100µ	0,701	0,389	0,28	2	0,70 €	3,89 €	56,00 €
	Diode:	SS12		0,0708	0,051	2	0,71 €	1,42 €	10,20 €
		SS54		0,241	0,182	1	2,41 €	2,41 €	18,20 €
	Inductor:	47µH	0,825	0,59	0,487	1	0,83 €	5,90 €	48,70 €
	Resistor:	2k		0,028	0,022	1	0,28 €	0,28 €	2,20 €
	Transistor:	SI2307		0,467	0,309	1	2,34 €	4,67 €	30,90 €
	Terminals:	1x2		0,567	0,502	1	5,67 €	5,67 €	50,20 €
	Regulator:	**LM2596-5.0**	4,4	3,55	2,91	1	4,40 €	35,50 €	291,00 €
	Resistor:	0k		0,016	0,01	1	0,16 €	0,16 €	1,00 €
						+		+	+
						Total:	17,49 €	59,90 €	508,40 €
						PPU:	17,49 €	5,99 €	5,08 €
	Regulator:	**LM2596-ADJ**	1,83	1,83	1,26	1	1,83 €	18,30 €	126,00 €
	Resistors:	1k		0,0112	0,0067	1	0,11 €	0,11 €	0,67 €
		3k		0,0117	0,007	1	0,12 €	0,12 €	0,70 €
						+		+	+
						Total:	14,99 €	42,77 €	343,77 €
						PPU:	14,99 €	4,28 €	3,44 €

Appendix 4:

Code: hd44780_settings.h (partially)

```
#define LCD_DB4_PORT             PORTB
// If using 4 bit mode, you must configure DB4-DB7
   #define LCD_DB4_PIN              0
   #define LCD_DB5_PORT             PORTB
   #define LCD_DB5_PIN              1
   #define LCD_DB6_PORT             PORTB
   #define LCD_DB6_PIN              4
   #define LCD_DB7_PORT             PORTB
   #define LCD_DB7_PIN              5

   #define LCD_RS_PORT              PORTB
// Port for RS line
   #define LCD_RS_PIN               3
// Pin for RS line

   #define LCD_RW_PORT              PORTB
// Port for RW line (ONLY used if RW_LINE_IMPLEMENTED=1)
   #define LCD_RW_PIN               2
// Pin for RW line (ONLY used if RW_LINE_IMPLEMENTED=1)

   #define LCD_DISPLAYS             1
// Up to 4 LCD displays can be used at one time

// All pins are shared between displays except for the E
// pin which each display will have its own

// Display 1 Settings - if you only have 1 display, YOU MUST
SET THESE
   #define LCD_DISPLAY_LINES        2
// Number of Lines, Only Used for Set I/O Mode Command
   #define LCD_E_PORT               PORTB
// Port for E line
   #define LCD_E_PIN                2
// Pin for E line
```

(Microchip Technology 2011)

Code: main.c

```
#define F_CPU 2000000UL
#include <avr/io.h>
#include <util/delay.h>

#include "hd44780.h"

uint8_t brightness = 0;

void Init()
{
//LCD Contrast PWM
		//Set 8-bit fast PWM mode
		TCCR0A|=(1<<WGM00)|(1<<WGM01)|(1<<CS00)|(1<<COM0
A1);
		//Initialize port B as output
		DDRB|=(1 << PB6)|(1 << PB7);	//DDRB  |=  0xC0;
		//DDRx '1' is OUTPUT, set b7 and b6 as output

		PINB |= (1<<PINB6);   //Set   Pin   6   (background
lighting) HIGH

		TCCR0B = (1 << CS00);
}

void SetPWMOutput(uint8_t duty)
{
		OCR0A=duty;
}

int main(void)
{
		Init();

		while(1)
		{
				// Contrast
				SetPWMOutput(25);

				//Drive Display
				lcd_init();
				lcd_clrscr();
				lcd_puts("HAMK Embedded");
				_delay_ms(1000);
		}

}
```

Auteursrechtelijke overeenkomst

Ik/wij verlenen het wereldwijde auteursrecht voor de ingediende eindverhandeling:
Board design and implementation for 8-bit avr microcontrollers in an educational environment

Richting: **master in de industriële wetenschappen: elektronica-ICT**
Jaar: **2018**

in alle mogelijke mediaformaten, - bestaande en in de toekomst te ontwikkelen - , aan de Universiteit Hasselt.

Niet tegenstaand deze toekenning van het auteursrecht aan de Universiteit Hasselt behoud ik als auteur het recht om de eindverhandeling, - in zijn geheel of gedeeltelijk -, vrij te reproduceren, (her)publiceren of distribueren zonder de toelating te moeten verkrijgen van de Universiteit Hasselt.

Ik bevestig dat de eindverhandeling mijn origineel werk is, en dat ik het recht heb om de rechten te verlenen die in deze overeenkomst worden beschreven. Ik verklaar tevens dat de eindverhandeling, naar mijn weten, het auteursrecht van anderen niet overtreedt.

Ik verklaar tevens dat ik voor het materiaal in de eindverhandeling dat beschermd wordt door het auteursrecht, de nodige toelatingen heb verkregen zodat ik deze ook aan de Universiteit Hasselt kan overdragen en dat dit duidelijk in de tekst en inhoud van de eindverhandeling werd genotificeerd.

Universiteit Hasselt zal mij als auteur(s) van de eindverhandeling identificeren en zal geen wijzigingen aanbrengen aan de eindverhandeling, uitgezonderd deze toegelaten door deze overeenkomst.

Voor akkoord,

van Roon, Gideon

Datum: **12/06/2018**