

Swakeup

More Than A Simple Wakeup Light

Elmar van Rijnswou and Maximilian Stiefel

June 2, 2017

Uppsala University

Table Of Contents

1. Introduction
2. Hardware
3. Software
4. Status Quo and Outlook

Introduction

Poka-yoke

Who knows what Poka-yoke is?

Poka-yoke

Poka-yoke (Wikipedia)

Poka-yoke [poka joke] is a Japanese term that means "mistake-proofing" or "inadvertent error prevention".

Background



Figure 1: According to Sveriges Radio many Swedes suffer from the winter blues or seasonal affective disorder. Image Source: Visitsweden

System Requirements

- Wakeup light which is a part of the *IoT*

System Requirements

- Wakeup light which is a part of the *IoT*
- Swakeup → engl. "Swedish Wakeup Light"

System Requirements

- Wakeup light which is a part of the *IoT*
- Swakeup → engl. "Swedish Wakeup Light"
- Wakes up, displays time, weather, mails, facebook

System Requirements

- Wakeup light which is a part of the *IoT*
- Swakeup → engl. "Swedish Wakeup Light"
- Wakes up, displays time, weather, mails, facebook
- Smart, small, USB charger included

System Requirements

- Wakeup light which is a part of the *IoT*
- Swakeup → engl. "Swedish Wakeup Light"
- Wakes up, displays time, weather, mails, facebook
- Smart, small, USB charger included
- Spends happiness

System Overview

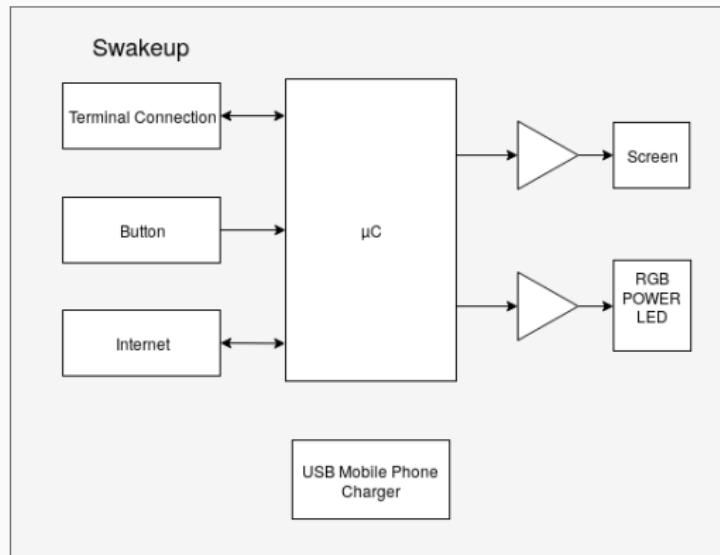


Figure 2: System Overview

System Overview

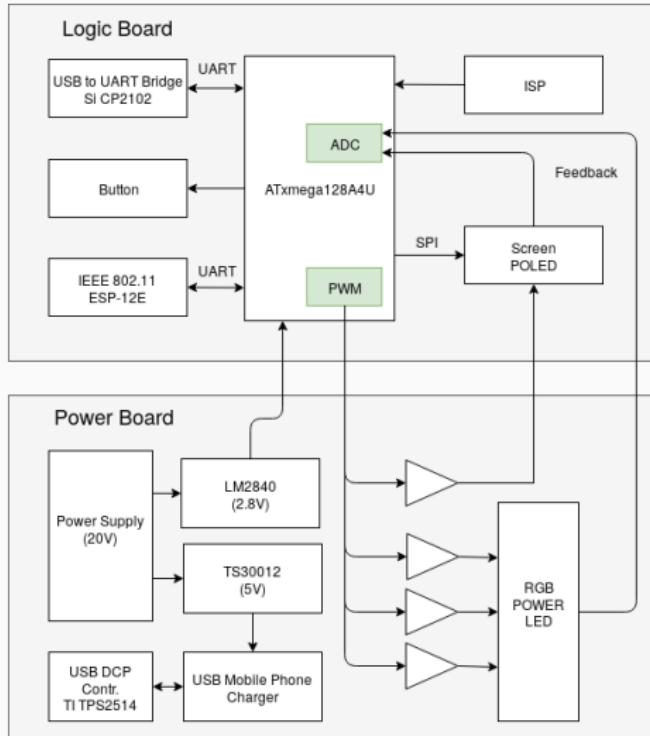


Figure 2: System Overview

Hardware

Power Board

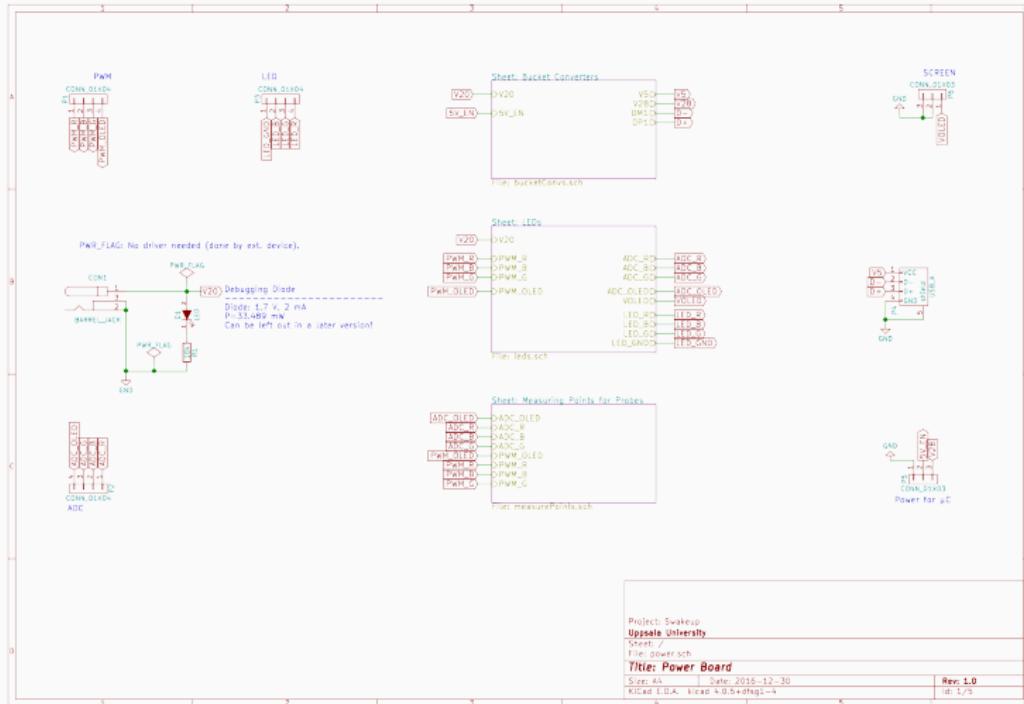


Figure 3: Power Board Schematics

Power Board

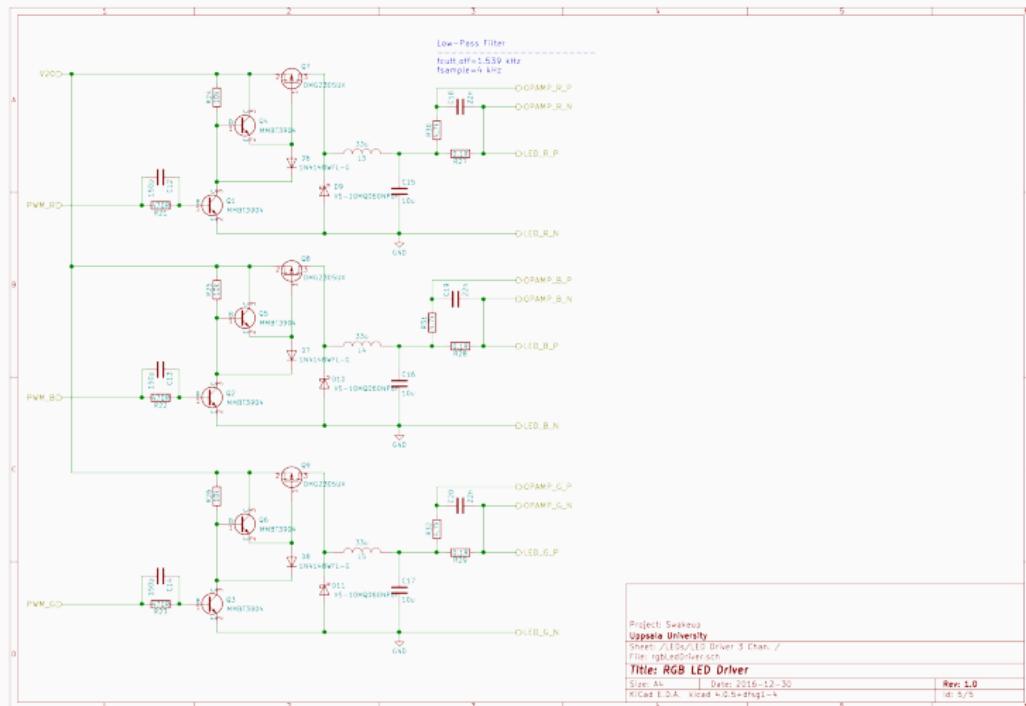


Figure 3: Power Board Schematics

Power Board

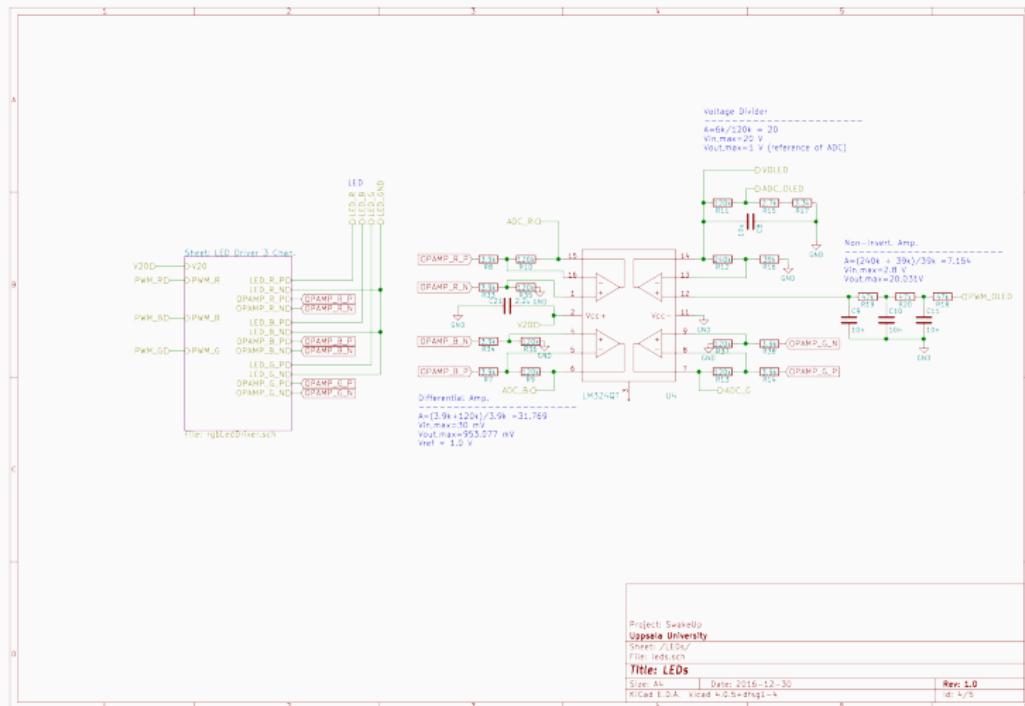
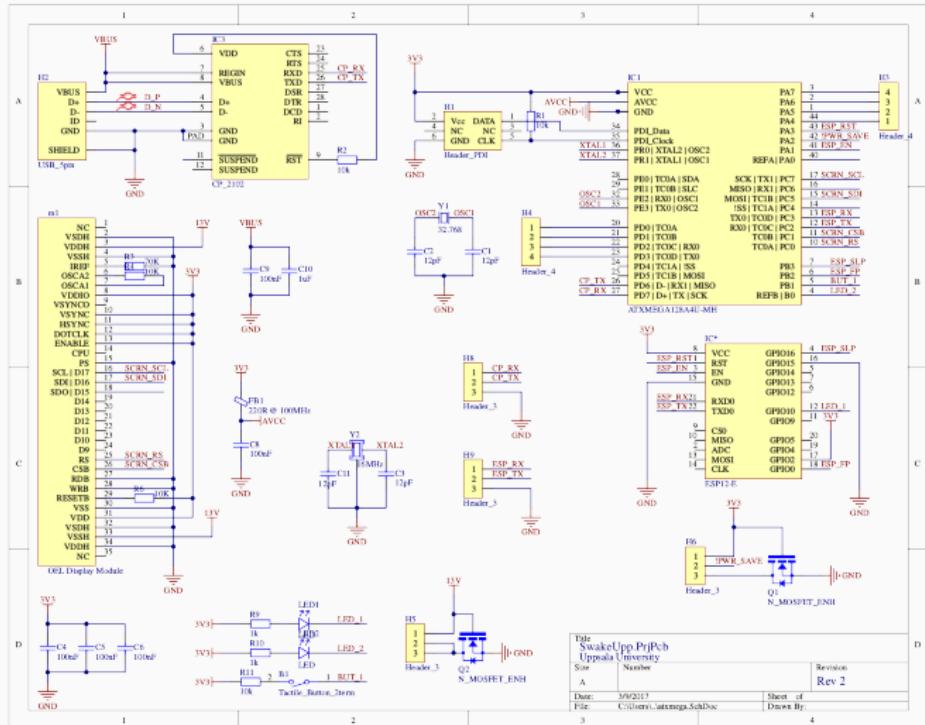


Figure 3: Power Board Schematics

Logic Board



Software

Code Structure

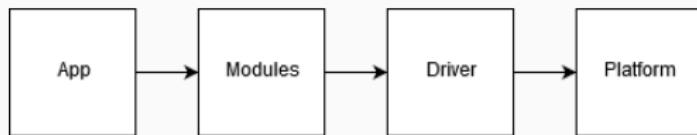


Figure 5: Abstract Layering Model

Code Organisation

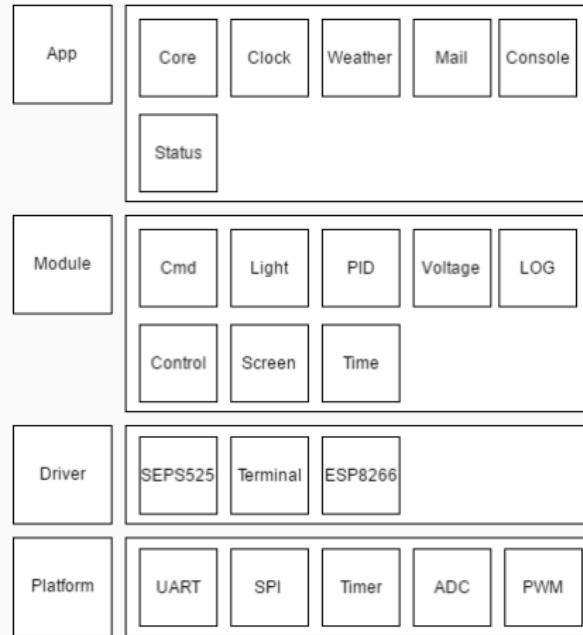


Figure 6: Block Diagram Of The Code Organisation

Operating System - Modules And Events

```
1 #define MODULE_DEFINE(VAR, DESC, INIT, DEINIT, ...)\n2     Module VAR = {\n3         .init = INIT,\n4         .deinit = DEINIT,\n5         .cnt = 0,\n6         .name = DESC,\n7         .deps = { __VA_ARGS__ } \n8     }\n9 MODULE_DEFINE(CORE, "Central core", init,_deinit, &TIME, &COMMAND, &ESP8266);
```

```
1 #define EVENT_REGISTER(eventName, desc)\\
2     Event eventName = \\n
3     {.eventId = __COUNTER__, .data = 0, .description = desc, .descLen = sizeof(desc) } \\
4 EVENT_REGISTER(EVENT_UART_DELIMITER, "Got UART delimiter");
```

```
1 event_addListener(&EVENT_UART_DELIMITER, callback);
```

```
1 event_fire(&EVENT_UART_DELIMITER, SYSTEM_ADDRESS_CAST (&delimiters[USART_ID][i]));
```

Drivers- ADC

```
1 //////////////////////////////////////////////////////////////////
2 // Macro: Create ISRs for the individual channels.
3 //////////////////////////////////////////////////////////////////
4 #define CREATE_ADCA_ISR(ADCVECTOR, VARSEMAPHORE, BUFFER, RESULTREG,STARTFLAG) \
5 ISR(ADCVECTOR) \
6 { \
7     static volatile uint8_t cycles = 0; \
8     if(!VARSEMAPHORE) \
9     { \
10         if( ((int16_t) RESULTREG) < 0) \
11             BUFFER[cycles++] = 0; \
12         else \
13             BUFFER[cycles++] = RESULTREG; \
14         if(cycles < NU_AVERAGING_VALS) \
15             ADCA.CTRLA |= STARTFLAG; \
16         else \
17         { \
18             VARSEMAPHORE= 1; \
19             cycles = 0; \
20         } \
21     } \
22 }
```

Realization (1)

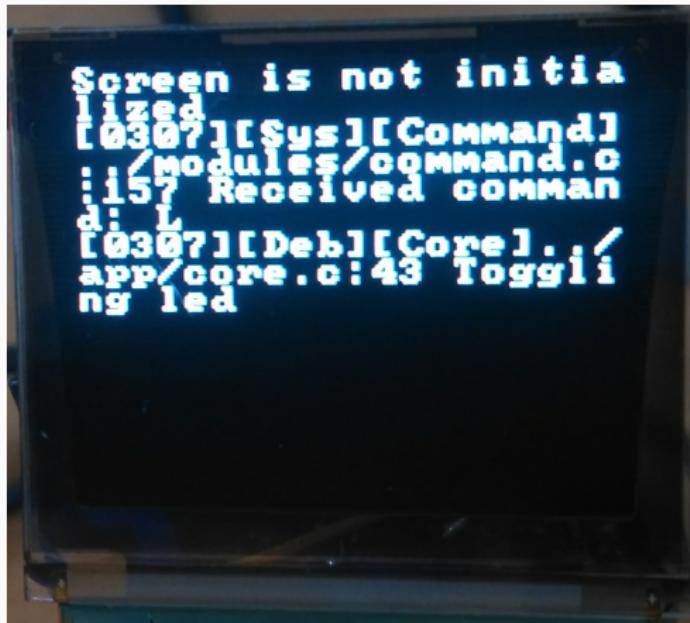


Figure 7: Screen Logging

Realization (2)

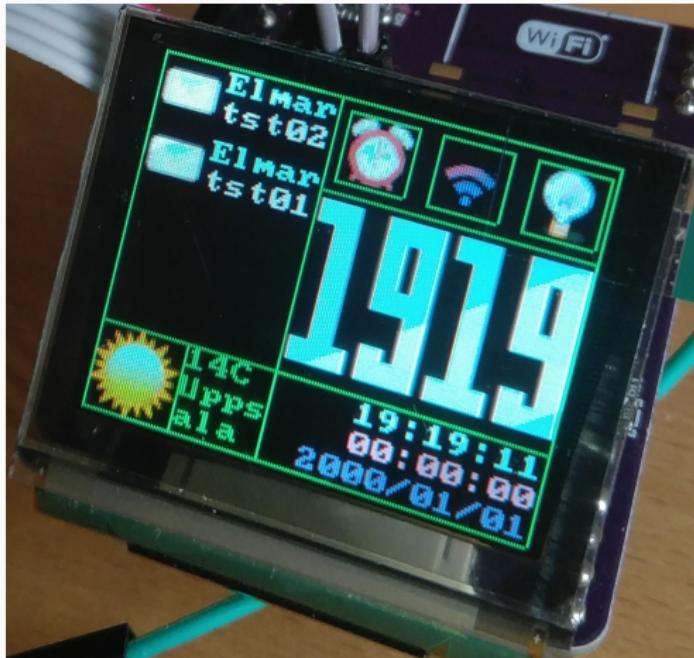


Figure 8: Appearance Of The Clock

Realization (3)

```
[0296][Sys][Command]../modules/command.c:132 Following commands are registered:  
? | Prints out this help  
A | Sends AT      A           no options  
G | Gets state of an app S<app>  
    W Get weather          no options  
    T Get time            no options  
L | Led control L<option> options: T(toggle) 1(on) 0(off)  
S | Sets an app state S<app> <options>  
    W<options> options: 1-6 for different weather  
    S<options> options: f(facebook) e(mail)  
    T<options> options: hour minute second  
T | Log sink     T<option> options: U(Uart) S(Screen)
```

Figure 9: USART Command Interpreter

Realization (4)

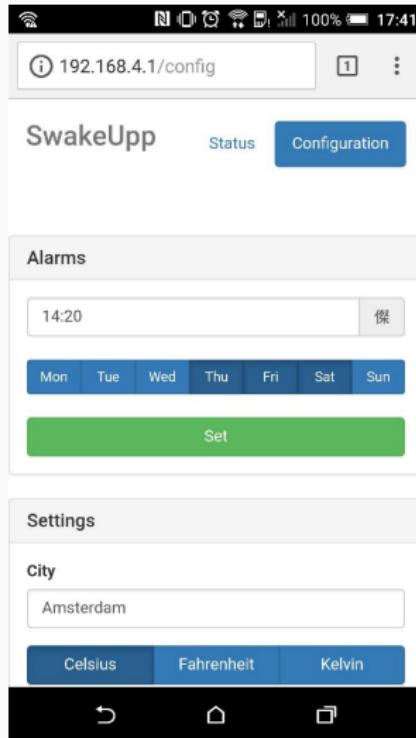


Figure 10: Website: Main User Interface

Realization (4)

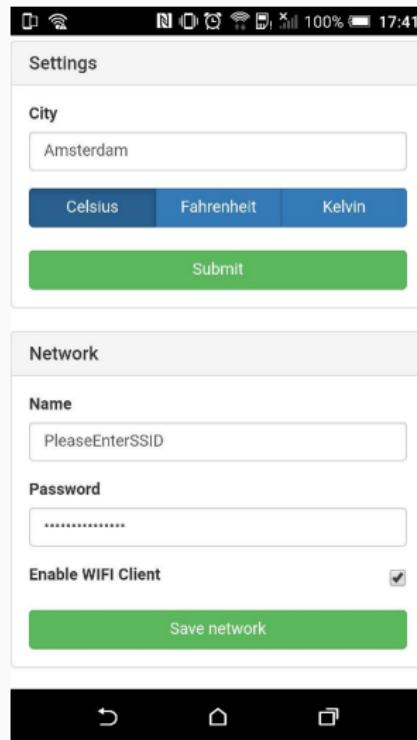


Figure 10: Website: Main User Interface

Realization (4)

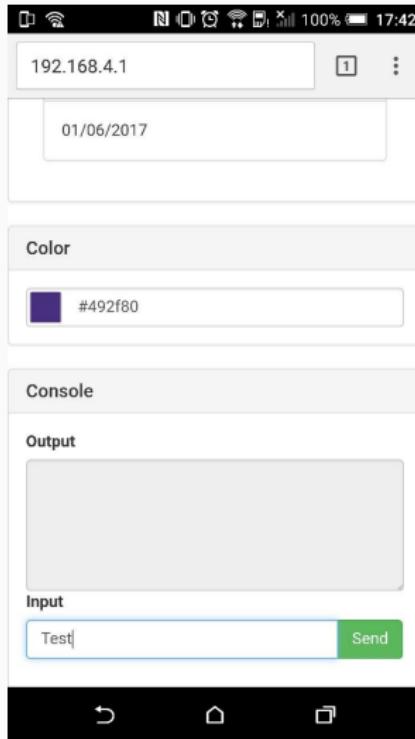


Figure 10: Website: Main User Interface

Status Quo and Outlook

Basic System Functionality is Given

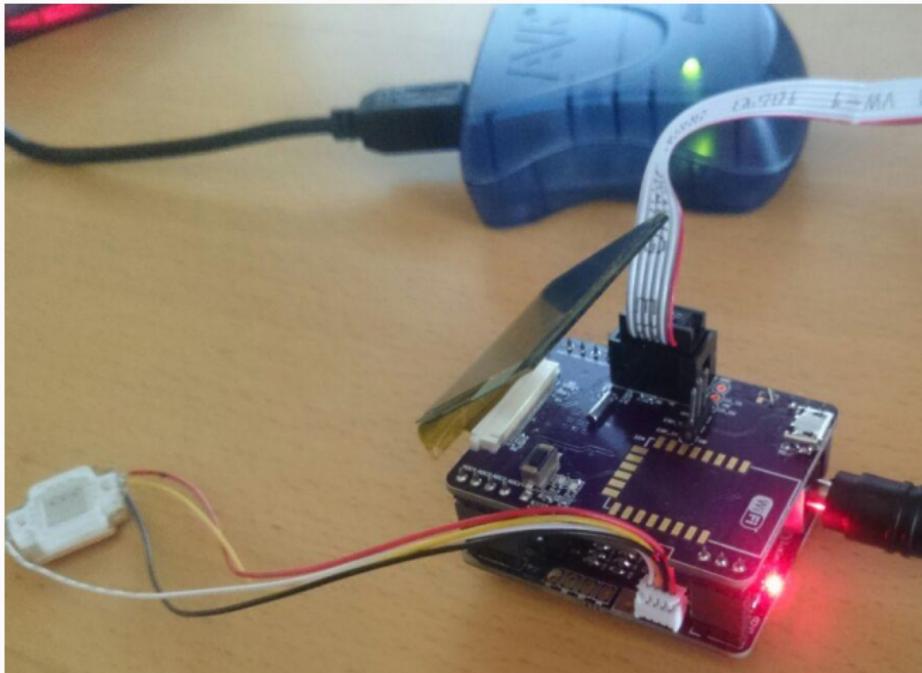


Figure 11: Picture of Nake HW

Hardware

HW Block	Working	Problem
USB Charging	✓	
Screen Driver	✓	
Screen Driver Feedback	✓	
Vcc for C	✓	
ESP	✓	
Terminal	✓	
LED Driver	✓	
LED Driver Feedback		Reversed inputs of the differential amplifier
USB DCP	✓	
Reverse Polarity Protection	✓	

Table 1: Status Quo Hardware

Software

SW Block	Working	Problem
UART	✓	
SPI	✓	
EPROM	✓	
Timer	✓	
ESP8266	✓	
Terminal	✓	
SEP525F	✓	
Wifi	✓	
Command	✓	
Log	✓	
Screen	✓	
Timekeeper	✓	
Controller		
Core	✓	
Weather	✓	
Clock	✓	
Social		Not implemented yet
PWM	✓	
ADC	✓	
Lightcontroller	✓	
PID	✓	
Advanced Light Patterns		Debugging still ongoing
Play 8 bit sounds		Not implemented, HW missing

Table 2: Status Quo Software

Outlook

- HW revision 3 is under construction

Outlook

- HW revision 3 is under construction
- Different functionality shall be added

Outlook

- HW revision 3 is under construction
- Different functionality shall be added
- Housing will be constructed

Outlook

- HW revision 3 is under construction
- Different functionality shall be added
- Housing will be constructed
- Social connectivity and calendar functions will be implemented

Outlook

- HW revision 3 is under construction
- Different functionality shall be added
- Housing will be constructed
- Social connectivity and calendar functions will be implemented
- Take part in Swedish Embedded Award (Embedded Priset)

Open Source

E: Elmar.Vanrijnswou.9818@student.uu.se

E: Maximilian.Stiefel.8233@student.uu.se



github.com/m3x1m0m/SwakeUp



Happy Coding :)