

Timebox 4 – Lock control

Oversigt

OpgaveNavn	Implementering af Funktionerne Verify key og Lock controller Opsætning af Embedded device og Lock systemet		
Implementering af krav	Hel implementering af kravene: EMB-2: "Skal kunne give signaler (0/1) til en elektronisk lås" EMB-3: "Skal kunne verificere en sendt "nøgle" " Delvis implementering af kravene: EMB-4: "Skal kunne afbryde forbindelsen ved forkert sendt "nøgle" " EMB-5: "Skal sende et signal til låsen efter 30 sek. at der skal låses."		
Udført af	Marc	Dato	15-10-2021
Timebox	4	Område	Embedded

Contents

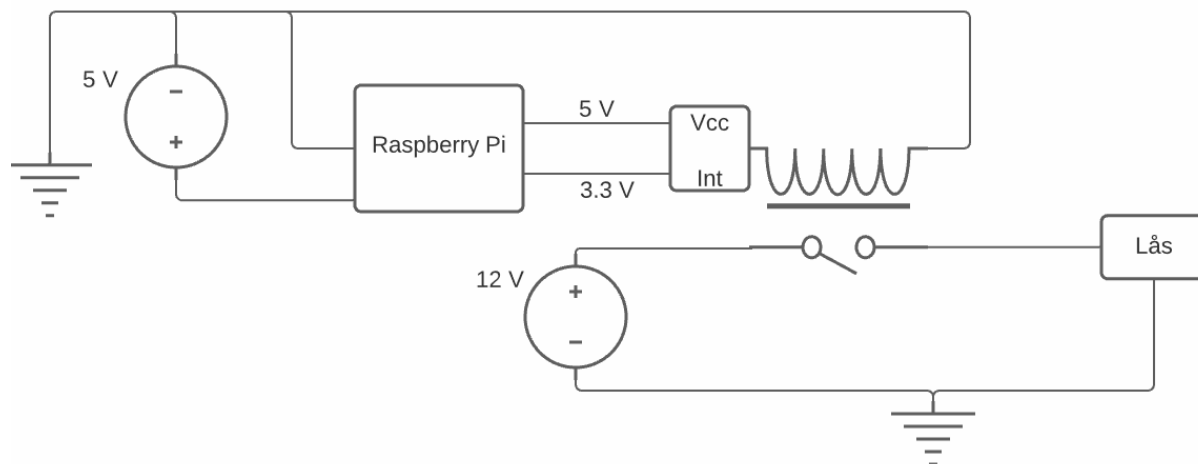
INTRODUKTION.....	1
ANALYSE OG DESIGN.....	2
IMPLEMENTERING	3
VERIFIKATION OG TESTRESULTAT	4
KONKLUSION	4
REFERENCER	5

Introduktion

Dette dokument omhandler at få sat Hardware delen af systemet op, alle komponenter blev købt ind efter "Launch Phasen", derfor mangles der blot at lave et diagram og opsætningen. Der blev yderlig implementeret to funktioner: "Verify key" som skal validere en modtaget nøgle og "Lock controller" som skal låse låsen op og låse låsen igen.


Analyse og Design

Proof of Concept som på Figur 1, hvor Raspberry Pi (RP) har en forsyning på 5 V og sender 5 V forsyning ud til relæet, derudover går der en ledning mere mellem relæet og RP, som skal styre relæet. Når der er strøm i relæet, bliver låsen tilsluttet 12 V, som låser låsen op.



Figur 1 Proof of Concept af systemet

De pinouts der skal anvendes fra Raspberry Pi er som på Figur 2. Der skal anvendes en ground til fælles stel og en 5 V til at styrer relæet, som "lås/lås op" anvendes GPIO 23 på pin 16, det er den GPIO som skal sende signalet ind til relæet for at give strøm til låsen.

 Raspberry Pi Pinout	
3v3 Power	1
GPIO 2 (I2C1 SDA)	3
GPIO 3 (I2C1 SCL)	5
GPIO 4 (GPCLK0)	7
Ground	9
GPIO 17	11
GPIO 27	13
GPIO 22	15
3v3 Power	17
GPIO 10 (SPI0 MOSI)	19
GPIO 9 (SPI0 MISO)	21
GPIO 11 (SPI0 SCLK)	23
Ground	25
GPIO 0 (EEPROM SDA)	27
GPIO 5	29
GPIO 6	31
GPIO 13 (PWM1)	33
GPIO 19 (PCM FS)	35
GPIO 26	37
Ground	39
5v Power	2
5v Power	4
Ground	6
GPIO 14 (UART TX)	8
GPIO 15 (UART RX)	10
GPIO 18 (PCM CLK)	12
Ground	14
GPIO 23	16
GPIO 24	18
Ground	20
GPIO 25	22
GPIO 8 (SPI0 CE0)	24
GPIO 7 (SPI0 CE1)	26
GPIO 1 (EEPROM SCL)	28
Ground	30
GPIO 12 (PWM0)	32
Ground	34
GPIO 16	36
GPIO 20 (PCM DIN)	38
GPIO 21 (PCM DOUT)	40

Figur 2 RP pinouts

Der er blevet anvendt en guide til at få hul i gennem (Hello World) til en GPIO [1].

Implementering

Som på Figur 3, er der lavet en klasse som indeholder de funktionen, som skal kaldes fra Bluetooth programmet. Klassen indeholder to locals: "signal" som er nummeret på den GPIO som skal anvendes og "key_code" som er nøglen til at låse låsen op. Under test forsøg vil 42 blive anvendt, mere kompleks kode med encoding og decoding, kan implementeres hvis der er tid. "Init" sætter GPIO'en der anvendes op.

```
1  import RPi.GPIO as GPIO
2  import time
3
4  class lock_control(object):
5      signal = 23
6      key_code = str(42)
7
8      def __init__(self):
9          GPIO.setmode(GPIO.BCM)
10         GPIO.setup(self.signal, GPIO.OUT)
11         GPIO.output(self.signal, GPIO.HIGH)
12
13     def validate_key(self, msg):
14         if msg == self.key_code:
15             bol = True
16         else:
17             bol = False
18         return bol
```

Figur 3 Class lock_control med funktionerne init og validate_key

De sidste to funktioner i klassen er som på Figur 4, hvor "unlock_lock" sender det signal ud der låser låsen op, og "lock_lock" sender det signal ud, som skal låse låsen.

```
21     def unlock_lock(self):
22         GPIO.output(self.signal, GPIO.LOW)
23         if GPIO.input(self.signal) == 0:
24             bol = True
25         else:
26             bol = False
27         return bol
28
29     def lock_lock(self):
30         GPIO.output(self.signal, GPIO.HIGH)
31         if GPIO.input(self.signal) == 1:
32             bol = True
33         else:
34             bol = False
35         return bol
36
```

Figur 4 Class lock_control med funktionerne unlock_lock og lock_lock

Selve "Proof of concept" sæt oppet er som på Figur 5, hvor den hvide boks er RP, den sort boks er batteriholderen, som leverer 12 V, kredsen i midten er relæet og yderst er selve låsen.



Figur 5 Proof of Concept sæt oppet

Verifikation og Testresultat

Der er ingen verifikation eller testresultat i dette dokument. Der henvises til TB4-BT server

Konklusion

Hardwaredelen af Proof of Concept systemet er blevet sat op.

Der er blevet implementeret en klasse, som har tre funktioner "Verify key" som skal validere en modtaget nøgle, "unlock_lock" som sender det signal ud, der skal låse låsen op og "lock_lock" der sender det signal ud, som skal låse låsen.

Det er testet i dokumentet "TB4-BT server".

Referencer

[1]	Raspberry gpio python https://sourceforge.net/p/raspberry-gpio-python/wiki/BasicUsage/ (sidst besøgt 15-10-2021)
-----	--