Mark Pedersen & Mads Finseth 17-06-2020

Computerteknologi HF1

LED Blink:

I dette eksempel har vi sat en LED til GrovePi, og med koden nedenunder, fortæller man at LED’en skal tænde, så skal den blive tændt i 1 sec, der efter skal den slukke i 1 sec, og sådan køre den i loop. Ved keyboard input slukker den.

```
import time
from Grovepi import *

# Connect the Grove LED to digital port D4
led = 4

pinMode(led,"OUTPUT")
time.sleep(1)

print ("This example will blink a Grove LED connected to the GrovePi+ on the port labeled D4.\nIf
you're having trouble seeing the LED blink, be sure to check the LED connection and the port
number.\n You may also try reversing the direction of the LED on the sensor.")
print (" ")
print ("Connect the LED to the port label D4!" )

while True:
        try:
        #Blink the LED
        digitalWrite(led,1)        # Send HIGH to switch on LED
        print ("LED ON!")
        time.sleep(1)

        digitalWrite(led,0)        # Send LOW to switch off LED
        print ("LED OFF!")
        time.sleep(1)

        except KeyboardInterrupt:    # Turn LED off before stopping
        digitalWrite(led,0)
        break
        except IOError:                       # Print "Error" if communication error encountered
        print ("Error")
```

LED Fade:

I dette eksempel bruger vi en LED og en Rotary Angle Sensor (potentiometer). I koden nedenfor fortæller man at led skal læse resistansen på potentiometeret, og på den måde styre hvor meget lys der skal sendes igennem LED’en. Når vi skruer op, lyser den mere og mere, og skruer vi helt ned vil den slukke.

```
import time
import grovepi

# Connect the Rotary Angle Sensor to analog port A2
potentiometer = 1

# Connect the LED to digital port D5
led = 3

grovepi.pinMode(led,"OUTPUT")
time.sleep(1)
i = 0

while True:
        try:
        # Read resistance from Potentiometer
        i = grovepi.analogRead(potentiometer)
        print(i)

        # Send PWM signal to LED
        grovepi.analogWrite(led,i//4)

        except IOError:
        print("Error")
```

Button buzzer:

I dette eksempel har vi sat en knap og en buzzer til henholdsvis port D2 og D4. Med koden under vil buzzeren ‘’ringe’’ når knappen bliver trykket. der vil blive printet på skærmen når vi tænder for buzzeren og printet off når vi slukker buzzeren. vi kan også stoppe buzzeren ved hjælp af keyboard.

```
import time
from grovepi import *
import math

buzzer_pin = 2              #Port for buzzer
button = 4          #Port for Button

pinMode(buzzer_pin,"OUTPUT")    # Assign mode for buzzer as output
pinMode(button,"INPUT")           # Assign mode for Button as input
while True:
    try:
            button_status= digitalRead(button)    #Read the Button status
            if button_status:    #If the Button is in HIGH position, run the program
                    digitalWrite(buzzer_pin,1)
                    # print "\tBuzzing"
            else:     #If Button is in Off position, print "Off" on the screen
                    digitalWrite(buzzer_pin,0)
                    # print "Off"
    except KeyboardInterrupt:    # Stop the buzzer before stopping
            digitalWrite(buzzer_pin,0)
            break
    except (IOError,TypeError) as e:
            print("Error")
```

Ultrasonic And Relay:

I dette eksempel vil vi bruge en Ultrasonic Ranger (sensor) til at tænde et relæet når der er bevægelse foran sensoren. Der vil være en afstand på 10 centimeter, så hvis man er 15 cm vil relæet ikke tænde.

```
from grovepi import *

# Connect the Grove Ultrasonic Ranger to digital port D4
# SIG,NC,VCC,GND

ultrasonic_ranger = 4
Relay_pin = 2

pinMode(Relay_pin,"OUTPUT")

while True:
        try:
        # Read distance value from Ultrasonic
        distant = ultrasonicRead(ultrasonic_ranger)
        print(distant,'cm')
        if distant <= 10:
        digitalWrite(Relay_pin,1)
        else:
        digitalWrite(Relay_pin,0)

        except TypeError:
        print("Error")
        except IOError:
        print("Error")
```

Home Weather Display:

I dette eksempel bliver der lavet en vejrstation med en DHT og en RGB LED skærm. DHT’en vil måle temperatur og fugtighed og sende informationen til skærmen. Bliver vist hver gang skærmen opdatere. (Skærmen vil blinke i de intervaller som updaten er sat til).

```
from grovepi import *
from grove_rgb_lcd import *
from time import sleep
from math import isnan

dht_sensor_port = 7 # connect the DHt sensor to port 7
dht_sensor_type = 0 # use 0 for the blue-colored sensor and 1 for the white-colored sensor

# set green as backlight color
# we need to do it just once
# setting the backlight color once reduces the amount of data transfer over the I2C line
setRGB(0,255,0)

while True:
    try:
            # get the temperature and Humidity from the DHT sensor
             [ temp,hum ] = dht(dht_sensor_port,dht_sensor_type)
             print("temp =", temp, "C\thumidity =", hum,"%")

             # check if we have nans
             # if so, then raise a type error exception
             if isnan(temp) is True or isnan(hum) is True:
                         raise TypeError('nan error')

             t = str(temp)
             h = str(hum)

            # instead of inserting a bunch of whitespace, we can just insert a \n
            # we're ensuring that if we get some strange strings on one line, the 2nd one won't be
affected
             setText_norefresh("Temp:" + t + "C\n" + "Humidity :" + h + "%")

    except (IOError, TypeError) as e:
             print(str(e))
             # and since we got a type error
             # then reset the LCD's text
             setText("")

    except KeyboardInterrupt as e:
             print(str(e))
             # since we're exiting the program
             # it's better to leave the LCD with a blank text
             setText("")
             break

    # wait some time before re-updating the LCD
    sleep(0.05)
```

Tweet the temperature:

```
import twitter
import time
import grovepi
import math

# Connections
sound_sensor = 0         # port A0
light_sensor = 1           # port A1
temperature_sensor = 2  # port D2
led = 3              # port D3

intro_str = "DI Lab's"

# Connect to Twitter
api = twitter.Api(
        consumer_key='YourKey',
        consumer_secret='YourKey',
        access_token_key='YourKey',
        access_token_secret='YourKey'
        )

grovepi.pinMode(led,"OUTPUT")
grovepi.analogWrite(led,255)  #turn led to max to show readiness


while True:


        # Error handling in case of problems communicating with the GrovePi
        try:

        # Get value from light sensor
        light_intensity = grovepi.analogRead(light_sensor)

        # Give PWM output to LED
        grovepi.analogWrite(led,light_intensity/4)

        # Get sound level
        sound_level = grovepi.analogRead(sound_sensor)

        time.sleep(0.5)

        # Get value from temperature sensor
        [t,h]=[0,0]
        [t,h] = grovepi.dht(temperature_sensor,0)

        # Post a tweet
        out_str ="%s Temp: %d C, Humidity: %d, Light: %d, Sound: %d"
%(intro_str,t,h,light_intensity/10,sound_level)
        print (out_str)
        api.PostUpdate(out_str)
        except IOError:
        print("Error")
```

```
except KeyboardInterrupt:
exit()
except Exception as e:
print("Duplicate Tweet or Twitter Refusal: {}".format(e))

time.sleep(60)
```

Grove Ultrasonic Ranger:

I dette eksempel vil vi bruger sensoren til at måle afstanden fra sensoren og til genstanden som bliver holdt over. Den vil printe hvor langt afstanden er, med koden nedenunder.

```
from grovepi import *

# Connect the Grove Ultrasonic Ranger to digital port D4
# SIG,NC,VCC,GND

ultrasonic_ranger = 4
Relay_pin = 2

pinMode(Relay_pin,"OUTPUT")

while True:
        try:
        # Read distance value from Ultrasonic
        distant = ultrasonicRead(ultrasonic_ranger)
        print(distant,'cm')
        if distant <= 10:
        digitalWrite(Relay_pin,1)
        else:
        digitalWrite(Relay_pin,0)

        except TypeError:
        print("Error")
        except IOError:
        print("Error")
```

Grove Relay:

I dette eksempel vil vi bruge koden til at tænde for et relæ i 5 sekunder og derefter slukke det i 5 sekunder. Ved keyboard input kan man stoppe processen

```
import time
import grovepi

# Connect the Grove Relay to digital port D4
# SIG,NC,VCC,GND
relay = 4

grovepi.pinMode(relay,"OUTPUT")

while True:
        try:
        # switch on for 5 seconds
        grovepi.digitalWrite(relay,1)
        print ("on")
        time.sleep(5)

        # switch off for 5 seconds
        grovepi.digitalWrite(relay,0)
        print ("off")
        time.sleep(5)

        except KeyboardInterrupt:
        grovepi.digitalWrite(relay,0)
        break
        except IOError:
        print ("Error")
```