```
//CoT
#include <CircusESP32Lib.h>
char ssid[] = "ssid";                                   //Navn på nettverket
char password[] = "Passord";                            //Passordet til ruteren
char roomToken1[] = "eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiI1MDI5In0.ShAebiVX7qYbDuN8vzleWmlWF9CgvsTR
Gjls6HByh8k";              //Token til soverom 1
char doorControlToken[] = "eyJhbGciOiJIUzI1NiJ9.eyJqdGkiOiI0OTQ1In0.D38ksENOh7rz5DODvLrx9aPemF39
WIVLNd6VniKaOyo";          //Token til brukeren til inngangspartiet
char server[] = "www.circusofthings.com";               //Hvor serveren er
char roomTempKey1[] = "23088";                          //Nøkkelen til signalet for romtemperaturen
char termostatKey1[] = "6074";                          //Nøkkelen for termostatsignalet
char roomStatusKey1[] = "15089";                        //Nøkkelen for å skjekke om person 1 er hjemme
char visitorKey1[] = "25932";                           //Nøkkelen for å skjekke om person 1 får besøkende
char callAllKey[] = "8894";                             //Nøkkelen for å skjekke om en person ringer på til alle
CircusESP32Lib circusESP32(server, ssid, password);     //Leser inn nettadressen til CoT, ruternavn og p
assord


//Misc
int personState = 1;
const int sleepButton = 12;
const int buzzerPin = 2;
#define MINUTES 60000


//Temperatur - Declaration
float tempValue;
float voltage;
float temp;
int termostat;
const int tempPin = 35;
const int heatLed = 15;
const int radiatorLed = 4;
unsigned long tempTime1 = 0;
unsigned long tempTime2 = 0;


//Besøkende - Declaration
int visitorState = 0;
int prevVisitorState = 0;
bool callAll = 0;
unsigned long visitorTime1 = 0;
unsigned long visitorTime2 = 0;


//Lyssensor - Declaration
const int ledPin = 23;
const int sensorPin = 34;
const int lightSwitch = 16;
int sensorValue;
int lightValue;
unsigned long lightTime1 = 0;
unsigned long lightTime2 = 0;
bool lightState = 1;
```

```
//Motor - Declaration
const int EN1 = 14;
const int IN1 = 25;
const int IN2 = 26;
const int autoButton = 32;
const int motorButton = 33;
const int autoLed = 18;
int motorSpeed;
bool autoButtonState = 1;
bool motorButtonState = 0;
bool servoButtonState = 0;


//Servo - Declaration
#include <ESP32Servo.h>         //Inkluderer et bibliotek som styrer servoen
const int servoPin = 27;
const int servoButton = 13;
bool windowState = 0;
Servo servo;                    //Initialiserer et servo-objekt som kalles "servo"



//Setup
//-------------------------------------------------------------------------------------------------------------------------------------
void setup() {
//Misc
  Serial.begin(9600);
  circusESP32.begin();              //Initialiserer oppkoblingen mot CoT
  analogReadResolution(10);         //Setter analogRead oppløsning til 10 bits
  pinMode(sleepButton, INPUT);
  pinMode(buzzerPin, OUTPUT);


//Temperatur - Setup
  pinMode(tempPin, INPUT);
  pinMode(heatLed, OUTPUT);
  pinMode(radiatorLed, OUTPUT);


//Lyssensor - Setup
  pinMode(ledPin, OUTPUT);
  pinMode(sensorPin, INPUT);
  pinMode(lightSwitch, INPUT);
  ledcSetup(4, 144, 8);             //Starter et PWM-signal på kanal 4, frekvensen er 144Hz og bitraten er 8
  ledcAttachPin(ledPin, 4);         //Fester ledPin til PWM-kanal 4


//Motor - Setup
  pinMode(EN1, OUTPUT);
  pinMode(IN1, OUTPUT);
  pinMode(IN2, OUTPUT);
  pinMode(autoButton, INPUT);
```

```
  pinMode(motorButton, INPUT);
  pinMode(autoLed, OUTPUT);
  ledcSetup(5, 30, 8);              //Starter et PWM-signal på kanal 5, frekvensen er 30Hz og bitraten er 8
  ledcAttachPin(EN1, 5);            //Fester EN1 til PWM-kanal 5


//Servo - Setup
  servo.attach(servoPin);           //Fester servoPin til servo-opjektet
  pinMode(servoButton, INPUT);
}



//Loop
//-----------------------------------------------------------------------------------------------------------------------------
void loop() {
//Temperatur - Loop
  tempTime1 = millis();
  if((tempTime1 - tempTime2) >= 10*MINUTES){      //Kjører funksjonen for temperatur hvert tiende minutt
    temperature();
  }


//Besøkende - Loop
  visitorTime1 = millis();
  if((visitorTime1 - visitorTime2) >= 1*MINUTES){     //Kjører funksjonen for besøkende hvert minutt
    visitorTime2 = visitorTime1;
    visitor();
  }


//Lyssensor - Loop
  if(digitalRead(lightSwitch) == 1){                          //Hvis lysbryterknappen blir trykt inn vil statusen på lyse endr
es (Skrus av eller på)
    delay(150);
    lightState = !lightState;
    Serial.println("lightState = " + String(lightState));
  }
  if(lightState == 0){
    ledcWrite(4, 0);                                    //Hvis lysbryter er av, så skrus lyset av
  }
  lightTime1 = millis();
  if(((lightTime1-lightTime2) >= 5000) && (lightState == 1)){     //Hvis lysbryteren er på så kjøres funksjonen for ly
set hvert femte sekund
    lightFunction();
  }


//Motor & Servo - Loop
  if(digitalRead(autoButton) == 1){
    delay(150);
    autoButtonState = !autoButtonState;       //Hvis auto-knappen blir trykt, endres tilstanden mellom 0 og 1 (av og på
)
  }
```

```
  if(digitalRead(motorButton) == 1){
    delay(150);
    motorButtonState = !motorButtonState;     //Hvis motorknappen blir trykt, endres tilstanden på motoren (av eller p
å)
    autoButtonState = 0;                      //Auto-modus blir frakoblet
  }
  if(digitalRead(servoButton) == 1){
    delay(150);
    servoButtonState = !servoButtonState;     //Hvis servoknappen blir trykt, endres tilstanden på motoren (av eller på
)
    autoButtonState = 0;                      //Auto-modus blir frakoblet
  }

  if(autoButtonState == 0){                   //Hvis auto-modus er av, kalles funksjonen for manuell kontroll av soveromm
et
    bedroomManuel();
  }
  if(autoButtonState == 1){                   //Hvis auto-modus er på, kalles funksjonen for automatisk kontroll av sovero
mmet
    bedroomAuto();
  }


//Sleep - Loop
  if((digitalRead(sleepButton) == HIGH) || (personState == 0)){     //Hvis sleepknappen blir trykt på, eller personen f
orlater huset, blir den egendefinerte funksjonen "sleep()" kjørt
    delay(150);
    sleep();
  }
}



//Functions
//-------------------------------------------------------------------------------------------------------------------
void temperature(){
  tempTime2 = tempTime1;
  tempValue = analogRead(tempPin);                          //Leser analogt signal inn fra pinnen til temperatursensoren
  voltage = ((tempValue+30)/1024)*3.3;                      //Avlest verdi blir konvertert til spenning
  temp = (voltage - 0.5) * 100;                            //Temperaturen blir regnet ut, ut i fra spenningen
  Serial.println("tempValue = " + String(tempValue));
  Serial.println("Temperature = " + String(temp) + " C");
  circusESP32.write(roomTempKey1, temp, roomToken1);           //Skriver temperaturen i rommmet til CoT
  termostat = circusESP32.read(termostatKey1, roomToken1);     //Leser av termostaten i CoT

  if(termostat == 0){                  //Hvis termostaten i rommet er satt til 0,
    digitalWrite(radiatorLed, LOW);    //er ovnen skrudd av,
    digitalWrite(heatLed, LOW);        //og varmen er også av
  }
  else if(temp < termostat){           //Hvis målt temperatur i rommet er mindre enn termostattemperaturen,
    digitalWrite(radiatorLed, HIGH);   //er ovnen skrudd på,
    digitalWrite(heatLed, HIGH);       //og varmen er også på
  }
  else if(temp >= termostat){          //Hvis målt temperatur i rommet er større eller lik termostattemperaturen,
```

```
    digitalWrite(radiatorLed, HIGH);      //er ovnen skrudd på,
    digitalWrite(heatLed, LOW);           //men varmen er av
  }

  personState = circusESP32.read(roomStatusKey1, doorControlToken);     //Sjekker om person 1 er hjemme.
  Serial.println("Room Status: " + String(personState));               //Printer statusen på person 1 i Serial Monitor
}


void visitor(){
  visitorState = circusESP32.read(visitorKey1, doorControlToken);     //Leser av om noen ringer på til person 1
  Serial.println("Visitors: " + String(visitorState));
  callAll = circusESP32.read(callAllKey, doorControlToken);           //Leser av om noen ringer på til alle

  if(visitorState > prevVisitorState){      //Betyr at person får en besøkende til
    for(int i=0; i<3; i++){                 //Blir kjørt en ringetone som varsler person 1
      tone(buzzerPin, 500);                 //En tone blir spilt på buzzeren med frekvens på 500Hz
      delay(200);
      tone(buzzerPin, 700);                 //En tone blir spilt på buzzeren med frekvens på 700Hz
      delay(200);
      tone(buzzerPin, 900);                 //En tone blir spilt på buzzeren med frekvens på 900Hz
      delay(200);
    }
    noTone(buzzerPin);                      //Skrur av buzzeren
    prevVisitorState = visitorState;        //Oppdaterer statusen for besøkende
  }
  if(visitorState < prevVisitorState){      //Betyr at en av de besøkende må dra fordi huset er fullt, og for å slippe inn n
oen andre besøkende
    for(int i=0; i<10; i++){                //Blir kjørt en rask varsletone for å si ifra
      tone(buzzerPin, 750);
      delay(50);
      noTone(buzzerPin);
      delay(50);
    }
    prevVisitorState = visitorState;        //Oppdaterer statusen for besøkende
  }
  if(callAll == 1){                         //Betyr at noen ringer på til alle
    for(int i=0; i<3; i++){                 //Blir kjørt en egen ringetone slik at man hører forskjell på om noen ringer på kun
 til deg
      tone(buzzerPin, 500);
      delay(500);
      noTone(buzzerPin);
      delay(100);
    }
    circusESP32.write(callAllKey, 0, doorControlToken);     //CoT signalet blir oppdatert å resatt til 0, slik at det ikke
 ringes på uendelig mange ganger
  }
}


void lightFunction(){
  sensorValue = analogRead(sensorPin);                      //Leser av verdien på lyssensoren
  Serial.println("Lys: " + String(sensorValue));
  lightValue = map(sensorValue, 0, 1023, 255, 0);           //Innlest verdi for lyset fra 0-1023 blir konvertert til et P
```

```
WM signal mellom 0-255 for LEDen
  ledcWrite(4, lightValue);                              //hvis sensorverdien er 0, er PWM-signalet 255, altså full styrke i
pærene
  lightTime2 = lightTime1;
}


void bedroomAuto(){
  motorSpeed = map(temp, 15, 60, 10, 30);          //Setter hastigheten på motoren i forhold til temperaturen i rommet
  motorButtonState = 0;                             //Skrur av "manuell" styring av motor
  servoButtonState = 0;                             //Skrur av "manuell" styring av servo
  digitalWrite(autoLed, HIGH);                        //Skrur på en LED for å vise at auto modus er på
  if(temp > 22){                                   //Hvis temperaturen overstiger 22 grader vil motoren begynne å kjøre
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    ledcWrite(5, motorSpeed);
    if((temp > 23) && (windowState == 0)){         //Hvis temperaturen stiger videre over 23 grader åpnes også luftev
induet
      openWindow();
    }
    else if((temp < 23) && (windowState == 1)){     //Hvis temperaturen synker under 23 grader, men er fortsatt over
22 grader, lukkes kun vinduet
      closeWindow();
    }
  }
  else if(temp < 22){                              //Hvis temperaturen er under 22 grader vil motoren ikke lenger kjøre
    digitalWrite(IN1, LOW);
    digitalWrite(IN2, LOW);
    if(windowState == 1){                           //Hvis vinduet er i åpen tilstand, vil dette lukkes
      closeWindow();
    }
  }
}


void bedroomManuel(){
  digitalWrite(autoLed, LOW);                    //LEDen som viser at auto-modus er på, blir slått av
  if(motorButtonState == 1){
    motorSpeed = map(temp, 15, 60, 10, 30);     //Angir hastigheten på takvifta ut i fra temperaturen på soverommet
    digitalWrite(IN1, HIGH);
    digitalWrite(IN2, LOW);
    ledcWrite(5, motorSpeed);
  }
  if(motorButtonState == 0){
    ledcWrite(5, 0);
  }
  if((servoButtonState == 1) && (windowState == 0)){      //Hvis vinduet er lukket, men at at knappen er trykt slik at
 det skal åpnes, åpnes vinduet
    openWindow();
  }
  if((servoButtonState == 0) && (windowState == 1)){      //Hvis vinduet er åpent, men at at knappen er trykt slik at
det skal lukkes, lukkes vinduet
    closeWindow();
  }
```

```
}


void openWindow(){
  Serial.println("Opening window");
  for(int i=0; i<80; i++){              //Kjører servoen sakte til 80 grader utslag
    servo.write(i);
    delay(20);
  }
  windowState = !windowState;           //Endrer statusen på vinduet til "åpent"
}


void closeWindow(){
  Serial.println("Closing window");     //Kjører servoen sakte tilbake til 0 fra 80 grader utslag
  for(int i=80; i>0; i--){
    servo.write(i);
    delay(20);
  }
  windowState = !windowState;     //Endrer statusen på vinduet til "lukket"
}


void sleep(){
  if(windowState == 1){                         //Hvis vinduet er åpent, lukkes det
    closeWindow();
  }
  Serial.println("Going to sleep");
  esp_sleep_enable_ext0_wakeup(GPIO_NUM_12, 1);     //Setter pin 12 som wake-up pin hvis signalet der blir HIG
H
  esp_deep_sleep_start();                       //Setter ESP32 i deep sleep modus
}
```