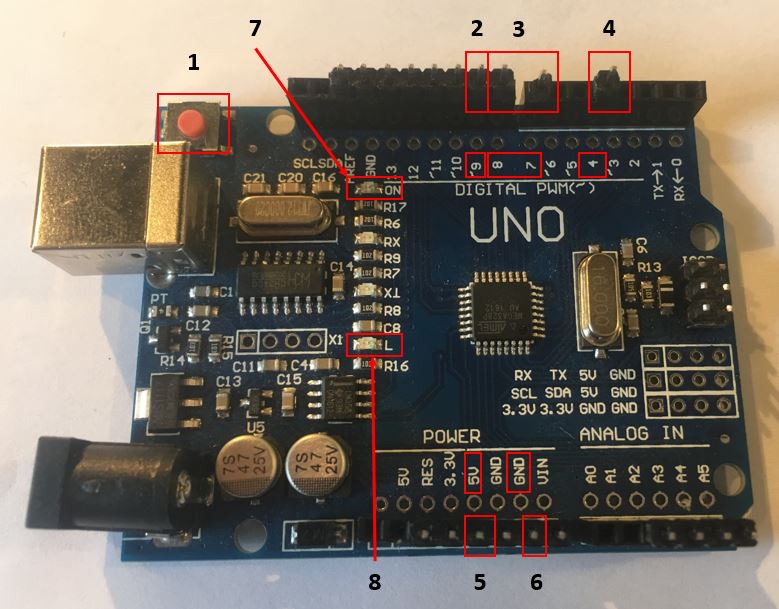
**Styrning av motorvärmare med GSM**

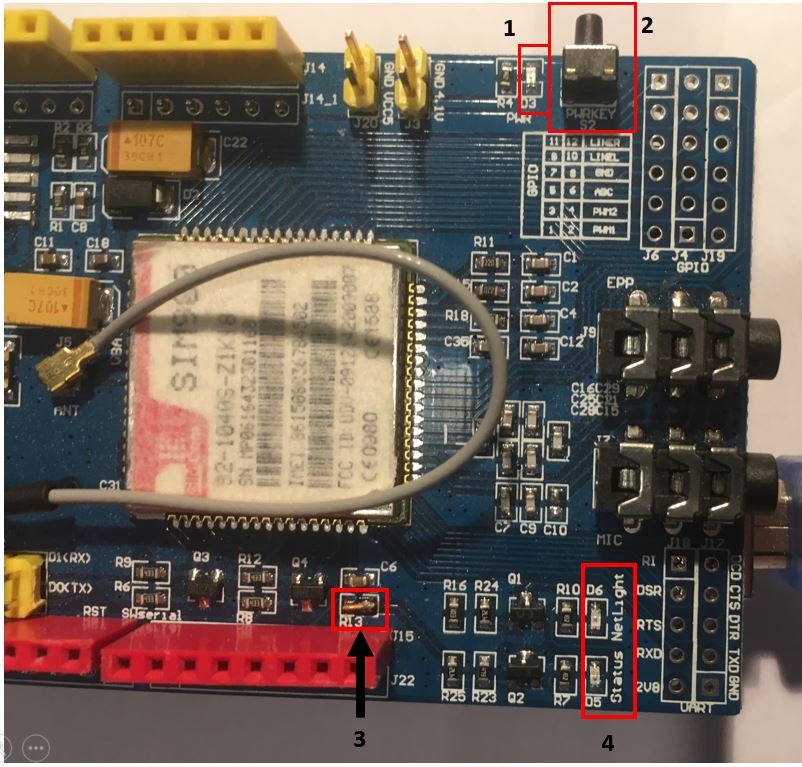
# Arduino



**Bild 1**

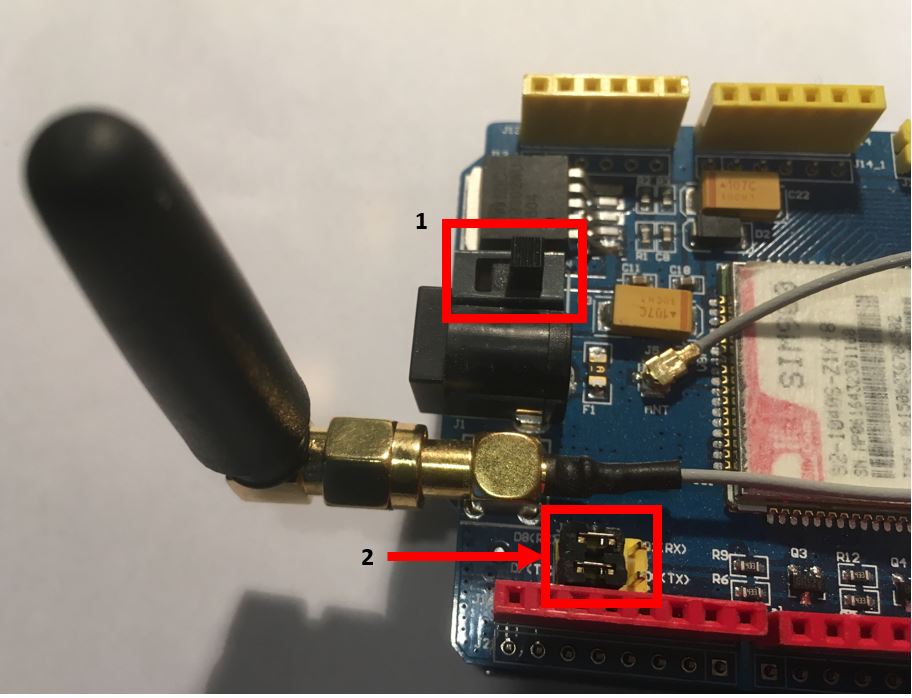
|  |  |
| --- | --- |
| Bild 1 | |
| 1 | Resetknapp för arduinon |
| 2 | Pin 9 används för att starta GSM via mjukvaran |
| 3 | Pin 8 Rx och Pin 7 Tx (Kommunikation mellan arduino och GSM) |
| 4 | Pin 4 i används för att slå av och på relä 4 på reläkortet |
| 5 | 5V till reläkortet |
| 6 | Jord till reläkortet |
| 7 | Diod ON lyser när arduinon är påslagen |
| 8 | Diod L blinkar vid uppstart och slocknar sedan. Blinkar när arduinon jobbar. |

# GSM



**Bild 2**

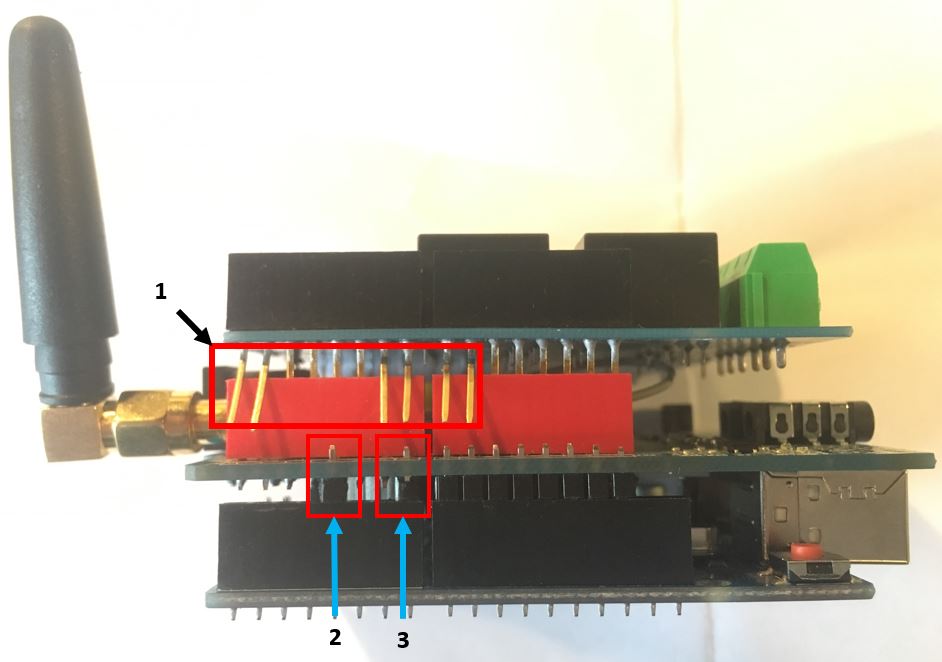
|  |  |
| --- | --- |
| Bild 2 | |
| 1 | Diod D3 lyser när GSM är strömsatt. |
| 2 | Av och på knapp för att koppla upp på nätet. Håll inne 1sek vid av och påslag. |
| 3 | Byglat R13 för att arduinon ska kunna slå av och på GSM:ens uppkopplin på nätet via Pin 9. |
| 4 | Led D5 ”Status” lyser när GSM:en är påslagen för att vara uppkopplad på nätet.  Led D6 ”NatLight” blinkar fort när den söker nät och blinkar sakta när den är uppkopplad på nätet. |



**Bild 3**

|  |  |
| --- | --- |
| Bild 3 | |
| 1 | Strömbrytaren visar det läge som gäller för extern spänningskälla. Arduinon klarar inte av att leverera den spänning som krävs för att GSM ska skicka sms. |
| 2 | 2st jumper/byglingar. Ska vara byglat enligt bilden för att arduinon ska kunna kommunicera med GSM:en via Pin 7 och Pin 8. |

# Hopkoppling



**Bild 4**

|  |  |
| --- | --- |
| Bild 4 | |
| 1 | De ben som är utvikta kan inte vara inkopplade för de stör GSM:en. |
| 2 | Pin 4 som används för styrning av relä 4. |
| 3 | Pin 7 Tx som används för kommunikation mellan arduino och GSM |
| Den rad med Pin 4 och Pin 7 är de enda som kan vara kopplade mellan arduino och GSM. Om fler pins på den raden kopplas samman med GSM kortet kommer GSM:en inte fungera. | |

# SMS kommandon

|  |  |
| --- | --- |
| OBS! Alla sms måste börja med versal. | |
| On30 | Slår på relä 4 i 30min och skickar bekräftelse ”Heater ON for 30min”. |
| On60 | Slår på relä 4 i 60min och skickar bekräftelse ”Heater ON for 60min”. |
| On120 | Slår på relä 4 i 120min och skickar bekräftelse ”Heater ON for 120min”. |
| Off | Slår av relä 4 oh skickar bekräftelse ”Heater is OFF”. |
| Status | 1. Om värmare är på:   * Returnerar hur många minuter man valt att värmaren ska gå samt hur många minuter den varit på. Exempel: ”Heater is ON for 30min. Heater has been ON for 12min.   2. Om värmare är avslagen:   * Returnerar ”Heater is OFF”. |
| Help | Skickar meny: ”On30 On60 On120 Off Status Help” |
| Res | Omstart av arduino och därmed GSM med mjukvara. |
| Om relät är påslaget och sms On30, On60 eller On120 skickas returneras sms ”Heater already ON” tillbaka. | |
| Om relät är avslaget och sms Off skickas returneras sms ”Heater already OFF” tillbaka. | |

# Kod



<https://www.arduino.cc/en/Main/Software>

//##############################################################

//# #

//# GLÖM INTE ATT LÄGGA IN TELEFONNUMMER LÄNGRE NER #

//# #

//##############################################################

//-------------------Inkuldering av bibliotek och variabler------------------------

#include <SoftwareSerial.h>

//Skapa software serial objekt för att kommunicera med SIM900

SoftwareSerial mySerial(7, 8); //SIM900 Tx och Rx är kopplade till Arduino pin 7 och 8

unsigned long time; //Används fär att holla koll på millis()

unsigned long timercontrol = 0; //Används i timeloopen i funktionen "void relay(int t)"

const long intervall30 = 1800000; //30min

const long intervall60 = 3600000; //60min

const long intervall120 = 7200000; //120min

int stat = 0; //Global variabel för sms funktionen Status. Används som parameter för att veta hur många minuter man valt att värmaren ska vara påslagen.

String uppstart = "A";//Variabel för att kontrollera att GSM startat

String compare = "A";//Variabel för att kontrollera att GSM startat

//--------------------------setup-------------------------

void setup() {

//Skickar en puls på pin 9 för att starta GSM

pinMode(9, OUTPUT);

digitalWrite(9, HIGH);

delay(1000);

digitalWrite(9, LOW);

delay(5000);

//Startar kommunikation med Arduino och Ardino IDE (seriell monitor)

Serial.begin(19200);

//Startar seriell kommunikation med Arduino och SIM900

mySerial.begin(19200);

// Serial.println("Startar upp..."); //Kod för monitorfönstret

delay(1000);

//#################################################################

//Kollar av så att GSM är uppstartad. Om inte skickar pin 9 en puls för att starta GSM.

mySerial.println("AT+CEER"); //AT+CEER är Extended Error Report kommando som anger GSM status

delay(100);

while (Serial.available()) {

mySerial.write(Serial.read());

}

while (mySerial.available()) {

compare = Serial.write(mySerial.read());

}

if (compare == uppstart) {

pinMode(9, OUTPUT);

digitalWrite(9, HIGH);

delay(1000);

digitalWrite(9, LOW);

delay(5000);

}

//################################################################

mySerial.println("AT"); //Arduinon skakar hand med SIM900

updateSerialSend();

mySerial.println("AT+CMGF=1"); //Ställer in formatet på sms. Inställningen 1 betyder TEXT mode.

updateSerialSend();

mySerial.println("AT+CNMI=1,2,0,0,0"); //Bestämmer hur mottagna sms ska hanteras

updateSerialSend();

mySerial.println("AT+CSCS=\"IRA\""); //Ställer in teckenkod IRA international reference alphabet (ITU-T T.50)

updateSerialSend();

//Kod för att kolla vilken teckentabell som är inställd

// mySerial.println("AT+CSCS?");

// updateSerialSend();

//Kod för att kolla vilka teckentabeller som finns i SIM900 kortet på GSM modulen

// mySerial.println("AT+CSCS=?");

// updateSerialSend();

// // Slår på funktionen att hämta tiden från nätet

// mySerial.println("AT+CLTS=1");

// updateSerialSend();

// // Sparar intällningar i GSM minne. OBS ej använd!!

// mySerial.println("AT&W");

// updateSerialSend();

// // Visar i serail monitor vad den inställda klockan är i GSM modulen

// mySerial.println("AT+CCLK?");

// updateSerialSend();

//####################################################

//Använder pin 4 för output till relä

//pin4 ger relä4

pinMode(4, OUTPUT); //relä 4

pinMode(LED\_BUILTIN, OUTPUT);

digitalWrite(LED\_BUILTIN, LOW); //Slår av diod L på Arduinokoret

}

//--------------------------Slut SETUP-----------------------------------------

//---------------------------Funktioner----------------------------------------

//####################################################################

//# #

//# I FUNKTIONEN void telnr() SKRIVS TELEFONNUMRET IN MED LANDSKOD #

//# #

//# EXEMPEL: +46123456789 (+46 ÄR SVERIGE) #

//# #

//####################################################################

void telnr() {

mySerial.println("AT+CMGS=\"+46XXXXXXXXX\"");

}

//###########################################################

//Stänger av GSM säkert och ser till så att GSM startar vid reset.

//Om inte GSM stängs av före arduinon kommer GSM att stängas av då arduinon startar om

//eftersom arduinon skickar pin9 HIGH och sedan LOW i SETUP för att starta GSM.

void pwrdown() {

mySerial.println("AT+CPOWD=1");

delay(5000);

}

//---------------Resetfunktion som resetar arduinon------------------------------

//############################################################

void(\* resetFunc) (void) = 0; //############

//############################################################

//-------------------------------------------------------------------------------

//##############################################################

//Funktion som sänder sms

void updateSerialSend() {

// Serial.XXXX är för komunikation med dator

// mySerial.XXXX är för kommunikation med GPS

delay(100);

while (Serial.available()) {

mySerial.write(Serial.read());//Forward what Serial received to Software Serial Port

}

while (mySerial.available()) {

Serial.write(mySerial.read());//Forward what Software Serial received to Serial Port

}

}

//#################################

//Denna funktion används när värmare inte är tillslagen. Tar emot sms och skickar vidare till funktionen: int evaluate(String in, int k)

void updateSerialRecieve() {

int k = 0;

String smsin;

if (mySerial.available() > 0) {

smsin = mySerial.readString();

}

evaluate(smsin, k);

}

//#############################

//Utvärdering av inkommet sms

int evaluate(String in, int k) {

int pin4 = digitalRead(4);

//Möjliga sms

String on30 = "On30";

String on60 = "On60";

String on120 = "On120";

String off = "Off";

String tid = "Status";

String help = "Help";

//###############################

//Resetkommando för att reseta arduino via sms. Används för att admin ska kunna starta om arduinon med sms. Reset kan bara göras då motorvärmare inte är på dvs pin4 är LOW.

String res = "Res";

if (in.indexOf(on30) > 0 && k == 0) {

// Serial.println("Motorvarmare på 30min");

stat = 30; //stat är global int variabel deklarerad innan SETUP

lista(1);

}

else if (in.indexOf(on30) > 0 | in.indexOf(on60) > 0 | in.indexOf(on120) > 0 && pin4 == HIGH) {

// Serial.println("Motorvarmare fortfarande på");

lista(3);

int u = 0;

return u;

}

else if (in.indexOf(off) > 0 && k == 1) {

// Serial.println("Motorvarmare slås av när den är på");

stat = 0;

int u = 1;

return u;

}

else if (in.indexOf(off) > 0 && pin4 == LOW) {

// Serial.println("Meddelande att motorvärmare redan är av när den får kommando att slå av.");

lista(4);

}

else if (in.indexOf(tid) > 0) {

int minut = (time - timercontrol) / 60000; //Beräknar hur många minuter värmaren varit på

String minuten;

String tiden;

minuten.concat(minut); //concat gör om int till String och lägger in värdet i strängen minuten

if (stat == 0) {

tiden = String("Heater is OFF");

}

else if (stat == 30) {

tiden = String("Heater ON for 30min.\nHeater has been ON for: " + minuten + "min");

}

else if (stat == 60) {

tiden = String("Heater ON for 60min.\nHeater has been ON for: " + minuten + "min");

}

else if (stat == 120) {

tiden = String("Heater ON for 120min.\nHeater has been ON for: " + minuten + "min");

}

telnr();

updateSerialSend();

mySerial.print(tiden);

updateSerialSend();

mySerial.write(26);

int u = 0;

return u;

}

else if (in.indexOf(on60) > 0 && k == 0) {

// Serial.println("Motorvarmare på 60min");

stat = 60;

lista(5);

}

else if (in.indexOf(on120) > 0 && k == 0) {

// Serial.println("Motorvarmare på 120min");

stat = 120;

lista(6);

}

else if (in.indexOf(help) > 0) {

// Serial.println("Hjälpmeny");

String menu = "On30\nOn60\nOn120\nOff\nStatus\nHelp";

telnr();

updateSerialSend();

mySerial.print(menu);

updateSerialSend();

mySerial.write(26);

}

else if (in.indexOf(res) > 0) {

// Serial.println("Reset");

if (k == 1) { //Om värmare är på k==1 skickas kommando att stänga av värmaren innan reset

stat = 0;

int u = 1;

return u;

}

pwrdown();

resetFunc();

}

}

//###############################

//Händelser beroende på vilket sms som inkommit

void lista(int i) {

//Funktionerna telnr() och updateSerialSend() förbereder för sändning av sms

telnr();

updateSerialSend();

if (i == 1) {

mySerial.print("Heater ON for 30min");

updateSerialSend();

mySerial.write(26);

relay(i); //Funktion som slår på relä4 och håller koll på tiden

}

else if (i == 2) {

mySerial.print("Heater is OFF");

updateSerialSend();

mySerial.write(26);

digitalWrite(4, LOW); //Slår av relä4

}

else if (i == 3) {

mySerial.print("Heater already ON");

updateSerialSend();

mySerial.write(26);

}

else if (i == 4) {

mySerial.print("Heater already OFF");

updateSerialSend();

mySerial.write(26);

digitalWrite(4, LOW); //Slår av relä4

}

else if (i == 5) {

mySerial.print("Heater ON for 60min");

updateSerialSend();

mySerial.write(26);

relay(i); //Funktion som slår på relä4 och håller koll på tiden

}

else if (i == 6) {

mySerial.print("Heater ON for 120min");

updateSerialSend();

mySerial.write(26);

relay(i); //Funktion som slår på relä4 och håller koll på tiden

}

}

//####################################

//Tidsstyrning av relä

void relay(int t) {

//---------t==1 är 30min--------------

if (t == 1) {

int e = 1;

int j = 0;

digitalWrite(4, HIGH);

timercontrol = millis();

do {

j = updateSerialRecieveLoop();

time = millis();

if (time - timercontrol >= intervall30 || j == 1) {

lista(2);

e = 0;

}

} while (e);

}

//------------t==5 är 60min---------------

else if (t == 5) {

int e = 1;

int j = 0;

digitalWrite(4, HIGH);

timercontrol = millis();

do {

j = updateSerialRecieveLoop();

time = millis();

if (time - timercontrol >= intervall60 || j == 1) {

lista(2);

e = 0;

}

} while (e);

}

//-----------------t==6 är 120min----------------

else if (t == 6) {

int e = 1;

int j = 0;

digitalWrite(4, HIGH);

timercontrol = millis();

do {

j = updateSerialRecieveLoop();

time = millis();

if (time - timercontrol >= intervall120 || j == 1) {

lista(2);

e = 0;

}

} while (e);

}

}

//######################

//Kollar sms för timeloopen i funktionen "void relay(int t)" när motorvärmare på

int updateSerialRecieveLoop() {

int k = 1;

int a = 0;

String smsin;

if (mySerial.available() > 0) {

smsin = mySerial.readString();

}

a = evaluate(smsin, k);

return a;

}

//------------------Huvudloop---------------------------

void loop() {

//Resetar arduinon när det är 8000000millisekunder/~133.33min eller mindre kvar på millis() tills den rullar över. millis() maxvärde=4,294,967,295.

if (millis() >= 4286967295) {

pwrdown();

resetFunc();

}

updateSerialRecieve(); // Kollar ifall den fått ett inkommande sms

}