

Arduino für FunkAmateure

Arduino Einführung Teil 7

Infrarot-Fernsteuerung mit AX-1838HS

Wie gehe ich am besten vor?

1. Was will ich machen?
2. Libraries einrichten
3. Bauteile
4. Schaltplan Versuche 1 bis 3
5. Tastencodes
6. Versuch 1: Sketch Tastencodes an seriellen Monitor
7. Versuch 2: Sketch Tastencodes an LCD
8. Versuch 3: Sketch LED ein/ausschalten
9. Library „IRremote“ Funktionen

Was will ich machen?

IR Versuch 1	Tastencode auf seriellem Monitor in HEX anzeigen Testen der IR-Fernbedienung zusammen mit dem IR-Empfänger
IR Versuch 2	Tastencode auf LCD in HEX anzeigen Testen der IR-Fernbedienung zusammen mit dem IR-Empfänger
IR Versuch 3	Mit den Tasten der Fernsteuerung eine LED ein/ausschalten Taste „1“ drücken, dann LED an Taste „0“ drücken, dann LED aus

Libraries einrichten

LCD in Betrieb nehmen

Library für LCD einrichten:

1. Arduino-IDE beenden!
2. Library herunterladen:
http://www.dfrobot.com/image/data/TOY0046/LiquidCrystal_I2Cv1-1.rar
3. Entpacken mit 7-Zip
4. Ordner „LiquidCrystal_dfrobot“ nach „...\\arduino\\libraries“ kopieren
5. Arduino-IDE öffnen

Infrarot-Empfänger AX-1838HS

Library für AX-1838HS einrichten:

1. Arduino-IDE beenden!
2. Bestehende Library „RobotIRremote“ aus Ordner „libraries“ entfernen
3. Library herunterladen: *<https://github.com/z3t0/Arduino-IRremote/releases>*
4. Entpacken und Ordner Umbenennen in „IRremote“
5. Ordner „IRremote “ nach „...\\arduino\\libraries“ kopieren
6. Arduino-IDE öffnen

Bauteile

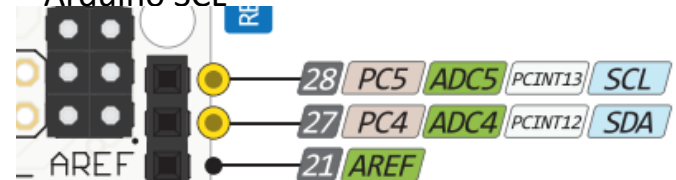
IR Versuch 1

ALLNET-Bausatz:
Steckbrett
ARDUINO
AX-1838HS
Fernsteuerung

IR Versuch 2

ALLNET-Bausatz: LCD
Anschlüsse:

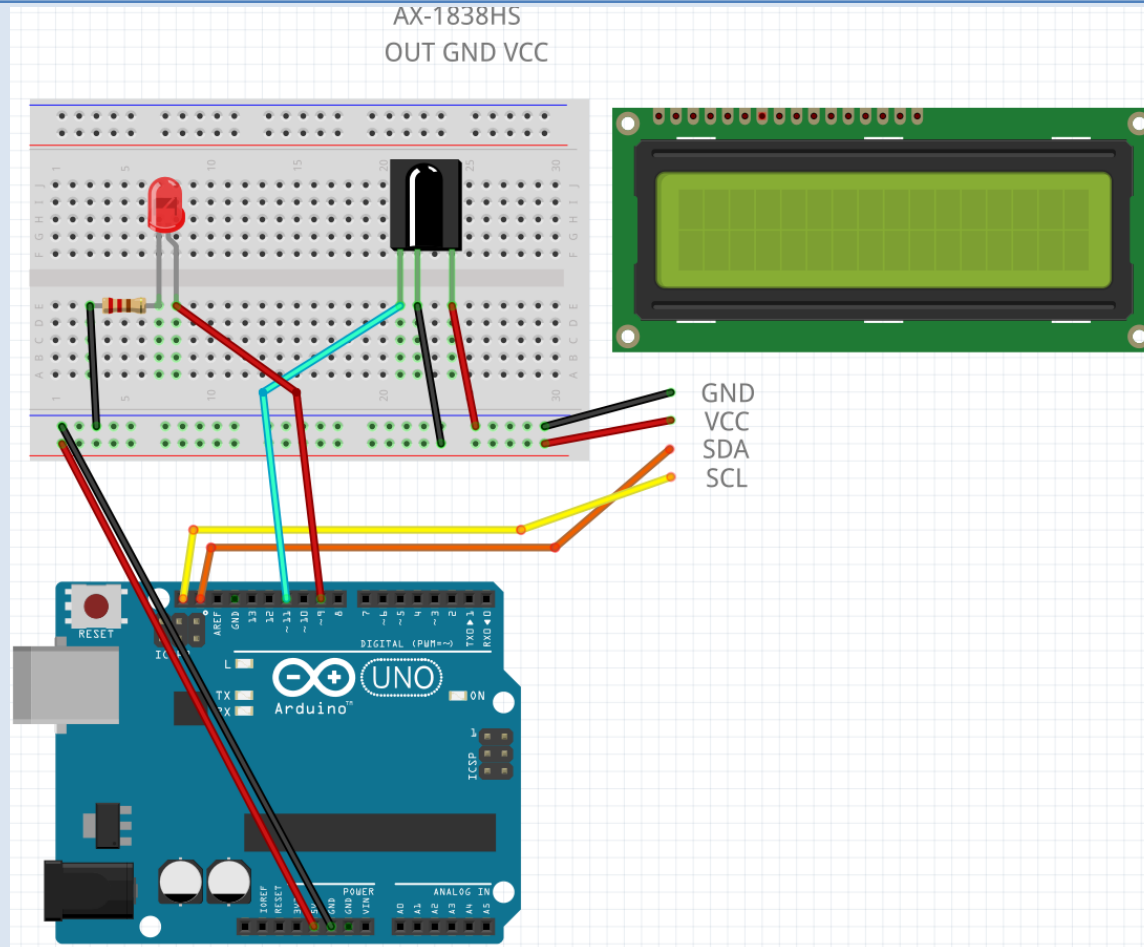
LCD	Arduino
GND	Steckbrett minus (-)
VCC	Steckbrett plus (+)
SDA	Arduino SDA
SCL	Arduino SCL



IR Versuch 3

LED
Vorwiderstand 220 Ohm

Schaltplan Infrarot-Empfänger AX-1838HS



Tastencodes

Codes der 21 Tasten
als HEX-Zahl

1= FFA25D;	2= FF629D;	3= FFE21D;
4= FF22DD;	5= FF02FD;	6= FFC23D;
7= FFE01F;	8= FFA857;	9= FF906F;
10= FF6897;	11= FF9867;	12= FFB04F;
13= FF30CF;	14= FF18E7;	15= FF7A85;
16= FF10EF;	17= FF38C7;	18= FF5AA5;
19= FF42BD;	20= FF4AB5;	21= FF52AD

Zahlensysteme Arduino-IDE

DEC (10er System)

BIN (2er System)

HEX (16er System)

Schreibweise im Programm:

143

0b10001111 oder B10001111

0x8F

Versuch 1: Sketch Tastencodes an seriellem Monitor

```
//ALL_Infrarot0.ino
#include <IRremote.h>

int RECV_PIN = 11;          // Signalleitung OUT muss mit Pin 11 verbunden
                             // sein
IRrecv irrecv(RECV_PIN);   // erzeuge ein Objekt von 'IRrecv'
decode_results results;    // erzeuge ein Objekt von 'decode_results'

void setup(){
  Serial.begin(9600);
  irrecv.enableIRIn();     // Empfänger starten
}
void loop() {
  if (irrecv.decode(&results)) { // haben wir ein IR Signal erhalten?
    long code = results.value;
    Serial.println(code, HEX);
    irrecv.resume();        // Empfänger zurücksetzen
  }
  delay(250);
}
```

Versuch 2: Sketch Tastencodes an LCD

```
// ALL_Infrarot1.ino
#include <IRremote.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

int RECV_PIN = 11;           // Signalleitung OUT muss mit Pin 11 verbunden sein
IRrecv irrecv(RECV_PIN);    // erzeuge ein Objekt von 'IRrecv'
decode_results results;     // erzeuge ein Objekt von 'decode_results'
LiquidCrystal_I2C lcd(0x27, 16, 2); // erzeuge ein Objekt von 'LiquidCrystal_I2C'

void setup(){
  lcd.begin();
  lcd.clear();
  irrecv.enableIRIn();      // Empfänger starten
}

void loop() {
  if (irrecv.decode(&results)) { // haben wir ein IR Signal erhalten?
    lcd.clear();
    long code = results.value;
    lcd.print(code, HEX);      // Ausgabe nicht in HEX
    irrecv.resume();          // Empfänger zurücksetzen
  }
  delay(250);
}
```


Versuch 3: Sketch LED ein/ausschalten

```
//ALL_Infrarot3.ino
#include <IRremote.h>
#include <Wire.h>
#include <LiquidCrystal_I2C.h>

int RECV_PIN = 11;           // Signalleitung OUT muss mit Pin 11 verbunden
                             // sein
IRrecv irrecv(RECV_PIN);    // erzeuge ein Objekt von 'IRrecv'
decode_results results;     // erzeuge ein Objekt von 'decode_results'
LiquidCrystal_I2C lcd(0x27, 16, 2); // erzeuge ein Objekt von 'LiquidCrystal_I2C'

long code;                  // Achtung HEX FF30CF verlangt long!
int ledPin=9;

void setup(){
  pinMode( ledPin,OUTPUT);
  lcd.begin();
  lcd.clear();
  lcd.print("LED ein/aus");
  lcd.setCursor(0,1);
  lcd.print("LED: aus");
  irrecv.enableIRIn();      // Empfänger starten
}
```

```
void loop() {
  if (irrecv.decode(&results) ) { // haben wir ein IR Signal erhalten?
    code = results.value;
    if ( code == 0xFF30CF ) {
      digitalWrite( ledPin, HIGH);
      lcd.setCursor(5,1);
      lcd.print("ein");
    }
    if ( code == 0xFF6897 ){
      digitalWrite( ledPin, LOW);
      lcd.setCursor(5,1);
      lcd.print("aus");
    }

    irrecv.resume();        // Empfänger zurücksetzen
  }
  delay(250);
}
```

Library „IRremote“ Funktionen

Receiver-Funktionen

Objekt „irrecv“ der Klasse „IRrecv“

irrecv.enableIRIn() Empfänger starten

irrecv.decode() Falls Code anliegt, dekodieren

irrecv.resume() Empfänger zurück setzen

Daten-Container

Objekt „results“ der Klasse decode_results

results.value Liefert den letzten dekodierten Code
