

Textul si imaginile din acest document sunt licentiate

Attribution-NonCommercial-NoDerivs
CC BY-NC-ND



Codul sursa din acest document este licentiat

Public-Domain

Esti liber sa distribui acest document prin orice mijloace consideri (email, publicare pe website / blog, printare, sau orice alt mijloc), atat timp cat nu aduci nici un fel de modificari acestuia. Codul sursa din acest document poate fi utilizat in orice fel de scop, de natura comerciala sau nu, fara nici un fel de limitari.

Encoder rotativ + Arduino

Despre encodere rotative

Encoderul este un dispozitiv electro-mecanic care convertește poziția unghiulară sau mișcarea unui arbore într-un semnal analogic sau digital. Există două tipuri de configurații mecanice pentru encodere optice: encodere rotative și encodere liniare.

Encoderele rotative optice sunt cel mai frecvent întâlnite în diverse sisteme de control în timp ce encoderele liniare sunt utilizate în special pentru aplicații de poziționare liniară.

Cu alte cuvinte, encoderul este folosit pentru a măsura cu precizie rotația motoarelor (viteză, nr. de pași) dar este utilizat și ca buton care se poate roti la infinit (fără cap de cursă ca la potentiometru). Există encodere echipate cu buton care se acționează atunci când apesi pe axul encoderului. Spre exemplu, un player MP3 poate fi echipat cu un encoder rotativ având un buton în centru. Atunci când roți encoderul poți schimba melodia sau volumul iar atunci când apesi butonul central (axul encoderului) poți opri/porni melodia.

Encoderele sunt caracterizate în special de rezoluție. Aceasta variază între 16 și 1024 de pași. Encoderele rotative se rotesc la infinit deoarece nu au capăt de cursă. Acesta este avantajul unui encoder în comparație cu un potentiometru dar este necesară memorarea numărului de pași pentru a cunoaște exact poziția.

Citeste mai multe detalii despre encodere:

<http://www.beisensors.com/technical-support-bei-rotary-encoder-vs-optical-encoder.html>

<http://www.beisensors.com/optical-rotary-encoders.html>

<http://www.tufts.edu/programs/mma/emid/RotaryEncoder.pdf>

<http://www.ni.com/white-paper/7109/en/>

<http://www.robofun.ro/forum>

Cum se conecteaza un encoder ?

Majoritatea encoderelor utilizeaza 3 pini, unul se conecteaza la GND iar ceilalti 2 la pinii placii Arduino sau orice alt microcontroller. Fiindca folosesti doar 2 pini conectati la placa, vei obtine in total 4 combinatii de stari si anume: 00, 01, 10, 11.

Sketch-ul Arduino monitorizeaza cele 4 combinatii de stari tinand cont de directia de rotatie si de numarul de pasi efectuati. Cu alte cuvinte, analizand combinatiile de stari placa Arduino detecteaza directia de rotatie si numarul de pasi parcursi de encoder. La prima vedere programul pare usor de implementat dar este necesara utilizarea sistemului de intreruperi.

Citeste mai multe detalii despre sistemul de intreruperi ale microcontrollerului ATmega:

<http://playground.arduino.cc/Code/Interrupts>

<http://bit.ly/1tyWDiF>

<http://bit.ly/XOksHL>

Conectarea encoderului la placa Arduino se realizeaza destul de simplu. Urmeaza tutorialul de mai jos:

<http://bildr.org/2012/08/rotary-encoder-arduino/>

Sketch-ul Arduino

Codul listat mai jos prindeaza prin Monitorul Serial numarul de pasi efectuati de encoder. Codul se incarca in placa Arduino si valorile se pot observa prin Monitorul Serial, linia Serial.println(encoderValue).

Daca encoderul se roteste intr-o directie atunci aceasta valoare poate sa creasca, linia:
if(sum == 0b1101 || sum == 0b0100 || sum == 0b0010 || sum == 0b1011) encoderValue ++;

Daca encoderul se roteste in directia opusa atunci valoarea printata scade, linia urmatoare:
if(sum == 0b1110 || sum == 0b0111 || sum == 0b0001 || sum == 0b1000) encoderValue --;

Observa setul de valori prin care se detecteaza directia de rotatie a encoderului. Codul tine evidenta starilor precedente, linia: lastEncoded = encoded;

<http://www.robofun.ro/forum>

```

//From bildr article: http://bildr.org/2012/08/rotary-encoder-
arduino/

//these pins can not be changed 2/3 are special pins
int encoderPin1 = 2;
int encoderPin2 = 3;

volatile int lastEncoded = 0;
volatile long encoderValue = 0;

long lastencoderValue = 0;

int lastMSB = 0;
int lastLSB = 0;

void setup() {
    Serial.begin (9600);

    pinMode(encoderPin1, INPUT);
    pinMode(encoderPin2, INPUT);

    digitalWrite(encoderPin1, HIGH); //turn pullup resistor on
    digitalWrite(encoderPin2, HIGH); //turn pullup resistor on

    //call updateEncoder() when any high/low changed seen
    //on interrupt 0 (pin 2), or interrupt 1 (pin 3)
    attachInterrupt(0, updateEncoder, CHANGE);
    attachInterrupt(1, updateEncoder, CHANGE);
}

void loop(){
    //Do stuff here

    Serial.println(encoderValue);
    delay(1000); //just here to slow down the output, and show it
will work even during a delay
}

void updateEncoder(){
    int MSB = digitalRead(encoderPin1); //MSB = most significant bit
    int LSB = digitalRead(encoderPin2); //LSB = least significant bit

    int encoded = (MSB << 1) | LSB; //converting the 2 pin value to
single number
    int sum = (lastEncoded << 2) | encoded; //adding it to the
previous encoded value

    if(sum == 0b1101 || sum == 0b0100 || sum == 0b0010 || sum ==
0b1011) encoderValue ++;
    if(sum == 0b1110 || sum == 0b0111 || sum == 0b0001 || sum ==
0b1000) encoderValue --;

```

```
    lastEncoded = encoded; //store this value for next time  
}
```

Exemple de aplicatii

- Lilypad MP3 Player: <http://bit.ly/1tPmIYR> sau <http://bit.ly/1qe6o3v>
- Controller de viteza pentru motoare DC cu Attiny 13: <http://bit.ly/1pbXHc3>
- Display cu LED-uri si encoder rotativ: <http://bit.ly/1su2LGf>
- Interfata intre PIC si encoder rotativ cu afisaj LCD: <http://bit.ly/1vJC9Dr>
- Controller de motoare stepper cu PIC si encoder rotativ: <http://bit.ly/1p6vwVH>