

P17 – Modulo Encoder

Encoder é um dispositivo eletromecânico que pode medir movimento ou posição. A maior parte dos encoders utiliza sensores ópticos para fornecer sinais elétricos no formato de trens de pulsos que podem, por sua vez, ser traduzidos em movimento, direção ou posição.

Os encoders rotatórios são usados para medir o movimento rotacional de um eixo. Mas esse tipo de encoder também é muito usado para ajuste de volume em aparelhos de som.



Especificacoes

Pulsos por revolução (Resolução) : 25;
Tensão de Alimentação: 5V;
Rotação: 360° (Sem limite);
2 Saídas de Pulsos defasadas 90°, possibilitando a leitura do sentido de giro. (Código Gray);
Dimensoes:
Largura: 17mm
Comprimento: 37mm

Sketch de exemplo

```
// -----  
// InterruptRotator.ino - Example for the RotaryEncoder library.  
// This class is implemented for use with the Arduino environment.  
// Copyright (c) by Matthias Hertel, http://www.mathertel.de  
// This work is licensed under a BSD style license. See http://www.mathertel.de/License.aspx  
// More information on: http://www.mathertel.de/Arduino  
// -----  
// 18.01.2014 created by Matthias Hertel  
// -----  
  
// This example checks the state of the rotary encoder in the loop() function.  
// The current position is printed on output when changed.  
  
// Hardware setup:  
// Attach a rotary encoder with output pins to A2 and A3.  
// The common contact should be attached to ground.  
  
#include <RotaryEncoder.h>
```

```
// Setup a RotaryEncoder for pins A2 and A3:
RotaryEncoder encoder(A2, A3);

void setup()
{
  Serial.begin(57600);
  Serial.println("SimplePollRotator example for the RotaryEncoder library.");

  // You may have to modify the next 2 lines if using other pins than A2 and A3
  PCICR |= (1 << PCIE1);    // This enables Pin Change Interrupt 1 that covers the Analog input
pins or Port C.
  PCMSK1 |= (1 << PCINT10) | (1 << PCINT11); // This enables the interrupt for pin 2 and 3 of
Port C.
} // setup()

// The Interrupt Service Routine for Pin Change Interrupt 1
// This routine will only be called on any signal change on A2 and A3: exactly where we need to
check.
ISR(PCINT1_vect) {
  encoder.tick(); // just call tick() to check the state.
}

// Read the current position of the encoder and print out when changed.
void loop()
{
  static int pos = 0;

  int newPos = encoder.getPosition();
  if (pos != newPos) {
    Serial.print(newPos);
    Serial.println();
    pos = newPos;

    // Just to show, that long lasting procedures don't break the rotary encoder:
    // When newPos is 66 the output will freeze, but the turned positions will be recognized even
when not polled.
    // The interrupt still works.
    // The output is correct 6.6 seconds later.
    if (newPos == 66)
      delay(6600);
  } // if
} // loop ()

// The End
```

E necessario a Biblioteca RotaryEncoder para funcionamento desta Sketch. Baixe e instale na pasta Libraries do Arduino.

Para download da Biblioteca RotaryEncoder clique no link disponivel em nossa pagina.
<http://www.gbkrobotics.com.br/produto/10/p17-encoder>