

Project: Connect atmega16 to rotation sensor
Hardware: Stk500, atmega16
Software: WinAvr (programmer's notepad), AVRStudio V.4.10

It is not essential that you use WinAvr but the guys who made this work are genius and since they deliver it for free, I myself feel I should use it. But every compiler you feel comfortable with will do (must be c compiler).

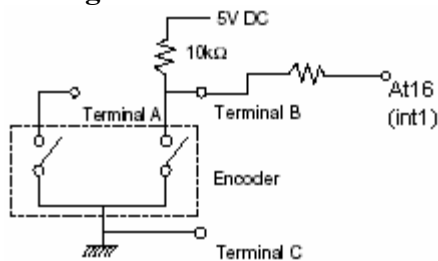
The project will do this:

A program which can handle the information send from the rotation sensor, and the speed calculated will decide which led will be lightening (if the speed is OK the two led in the middle will lit, but if there is to much speed then the led too the right will indicate that, and too little the led on the left side will indicate).

WinAvr

Know we need to make the program for the atmega16, and here we need timer (to calculate speed), external interrupt to check each pulse and we also need a little electrical circuit which operate the pulse generator.

Pulse generator



Program

The code is written with description all the way, and it should be enough to understand. But I will also set up a list over where to find the special things in the datasheet for the atmega16:

- External interrupt p.66
- Timers p.85

```
//-----  
// Project:                      Rotation sensor  
  
// Version:                      V1.01  
// Date:                        10 January 2005  
// Author:                      Roald  
// File:                        Rotation_sensor  
// Description:                 This will use Timer and external interrupt to build an
```

```

//                                     rotation sensor
//-----

//*****
//*****
// include files
//*****
//*****

// registers and interrupts definitions
#include <avr/io.h>                                     // include
library for microcontroller
#include <avr/interrupt.h>                             // include library
    for enable interrupt
#include <avr/signal.h>                                 // include
library for interrupt vector

//*****
//*****
// Definitions
//*****
//*****

#define Key (PINB & 0x01)

//*****
//*****
// function prototypes
//*****
//*****

void delay (int ms);                                  // declaration
for delay
void ShowResult();                                  //
declaration for ShowResult
void IntializeStep();                               // declaration
for IntializeStep

//*****
//*****
// constants
//*****
//*****

int const baudrate = 23;                            // sets baudrate to 9600

```

```

//*****
*****
// Global structs
//*****
*****

//*****
*****
// Global variables
//*****
*****

// int
int volatile AntStep ;
int volatile Timer;
int Seconds;
int AntStepOld;
int   AntStepNew;
int AntStepAv;
int AntSec;
int InitializeSet;

// double

// arrays

// chars

//*****
*****
// Interrupts
//*****
*****

// external interrupt
SIGNAL(SIG_INTERRUPT1)
{
    AntStep++;
}

// interrupt for transmitt
SIGNAL(SIG_UART_DATA)
{

```

```

}

// Timer interrupt
SIGNAL(SIG_OUTPUT_COMPARE2)
{
    if(Seconds++ == 0xE2)
    {
        Timer = 1;
        Seconds = 0;
    }
}

//-----*****-----
// Initialization
//-----*****-----
// Delay routine
void initialize (void)
{
    // ports
    DDRA = 0xFF; // portc
as out
    DDRB = 0x00; // portb
as in
    DDRD = 0x00; // portd
as in

    PORTA = 0xFF; // set default
porta

    //initialize external interrupt1
    MCUCR = (1<<ISC10)|(1<<ISC11);
    GICR = (1<<INT1);

    // Enable receiver and transmitter
    UCSRB = (1<<RXCIE)|(1<<TXCIE)|(1<<TXEN)|(1<<RXEN)|(1<<UDRIE);

    //UBRR = baudrate; // set
baudrate = 9600
    UBRRH = 0x00; // baudrate for atmega16
    UBRRL = baudrate;

    UCSRC = (1<<URSEL)|(1<<USBS)|(1<<UCSZ0)|(1<<UCSZ1);

    //timer interrupt

```

```

    TCCR2 = (1<<CS22);    // sets prescaling 64, lets the timer count each
second
    OCR2 = 0xff;         // sets the compare value
    TIMSK = (1<<OCIE2); // enable interrupts when compare is executed

    // enable global interrupts
    sei();

    //Initialize steps
    AntStep = 0;
    AntStepOld = 0;
    AntStepNew = 0;
    AntStepAv = 0;

    //initialize Timer
    Timer = 0;

    //initialize counter
    AntSec = 1;

    //initialize initializeSet
    InitializeSet = 0;
}

#####
// Functions
#####

// delay routine
void delay (int ms)
{
    int i,k,l;
    for (l=0; l<ms*5;l++)
    {
        for (i=0; i<199; i++)
            k++;
    }
}

// routine for showing the result, which will be done each second
void ShowResult()
{

```



```

// Main
//%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

void main(void)
{
    initialize ();
    while(1)
    {
        if(Key == 0 || (Timer == 1 && InitializeSet == 1))
        {
            InitializeSet = 1;           // variable to
check if it is in initialize

            // modus
            IntializeStep();           // jumps to subroutine
InitializeStep
        }
        else if(Timer == 1 && InitializeSet == 0) ShowResult();
    }
}

```