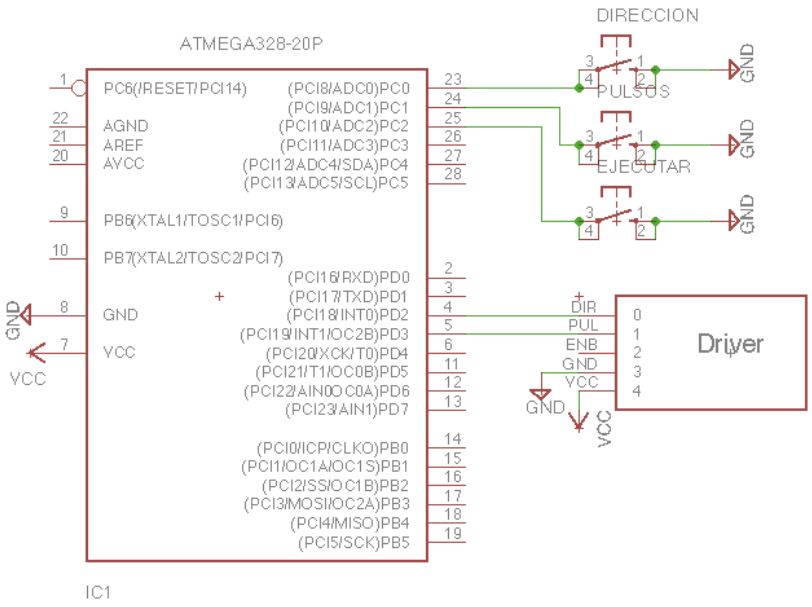
UNIVERSIDAD DE SANTIAGO DE CHILE
FACULTAD DE INGENIERÍA
Departamento de Ingeniería Mecánica
Área de Procesos Mecánicos

Control manual motor paso a paso

Se desea controlar un motor paso a paso de forma manual mediante 3 pulsadores

El switch “dirección” debe activar o desactivar el pin “DIR” del driver, ubicado en PD2, el switch “Pulsos” debe almacenar la cantidad de pulsos a ejecutar, aumentando 200 pulsos cada vez que se presiona, y el switch ejecutar debe realizar el número de pasos almacenado gracias al pulsador anterior, enviando los pulsos correspondientes al pin “PUL” del driver (1 pulso = HIGH-LOW)



Se pide:

- Utilizar procesos, donde se debe encontrar una fórmula que relacione la cantidad de RPM con la frecuencia del proceso.
- Configurar según sea necesario

Andres Utrera Soto

Solución:

```
/*
 *
 * Created: 17-02-2017 11:17:50
 * Author : Andres
 */
#include <avr/io.h>
#include <avr/interrupt.h>

typedef struct
{
	unsigned long TIMER;
	unsigned long DELTA;
	unsigned char IDTASK;
} PROC_STRUCT;

static volatile PROC_STRUCT Procesos[4];


#define Puerto_motor PORTD
#define DDR_motor DDRD
#define DIR PD2
#define PULSE PD3

#define PIN_botones PINC
#define Puerto_botones PORTC
#define boton_direccion PC0
#define boton_pasos PC1
#define boton_ejecutar PC2

#define PASOS 400//Pasos por revolucion
#define RPM 60


void setup_hardware()
{
	DDR_motor |= _BV(PULSE) |_BV(DIR);//Activa salidas

	Puerto_botones |= _BV(boton_direccion) |_BV(boton_pasos) |_BV(boton_ejecutar);//Activa pull up

}

void setup_timer()
{
	TCCR0B |= _BV(CS00);//Prescaler 1
	TIMSK0 |= _BV(TOIE0);//Timer 0 interrupt enable
	sei();//Activa interrupciones globales
}

void inicia_procesos()
{
	Procesos[0].IDTASK = 0;
	Procesos[0].TIMER = 0;
	Procesos[0].DELTA = 1875000/(PASOS*RPM);//asigna frecuencia para obtener RPM
	//Procesos[0].DELTA=30000;

	Procesos[1].IDTASK = 0;//Boton direccion
	Procesos[1].TIMER = 0;
	Procesos[1].DELTA = 800;//13ms aprox
```

```
	Procesos[2].IDTASK = 0;//Boton pasos
	Procesos[2].TIMER = 0;
	Procesos[2].DELTA = 800;

	Procesos[3].IDTASK = 0;//Boton ejecutar
	Procesos[3].TIMER = 0;
	Procesos[3].DELTA = 800;

}


ISR(TIMER0_OVF_vect)
{
	for(unsigned char i = 0;i<=3;i++)
	{
		Procesos[i].TIMER++;
	}


}


int main(void)
{
	setup_hardware();
	setup_timer();
	inicia_procesos();
	unsigned char direccion = 0;
	unsigned char deb_direccion = 0;
	unsigned long cont_pasos = 0;
	unsigned char deb_pasos=0;
	unsigned char ejecutar = 0;
	unsigned char deb_ejecutar =0;
    while (1)
    {
		if(ejecutar == 0)
		{
			if(bit_is_clear(PIN_botones,boton_direccion))
			{
				if(deb_direccion<= 250)
				{
					deb_direccion++;
				}
			} else {
				if(deb_direccion>=35)
				{
					//Hacer algo
					if(direccion==0)
					{
						direccion=1;
						Puerto_motor |= _BV(DIR);
					} else
					{
						direccion = 0;
						Puerto_motor &= ~_BV(DIR);
					}
				}
				deb_direccion=0;
			}
```

```
                if(bit_is_clear(PIN_botones,boton_pasos))
                {
                        if(deb_pasos<= 250)
                        {
                                deb_pasos++;
                        }
                } else {
                        if(deb_pasos>=35)
                        {
                                //Hacer algo
                                cont_pasos=cont_pasos+200;
                        }
                        deb_pasos=0;
                }


                if(bit_is_clear(PIN_botones,boton_ejecutar))
                {
                        if(deb_ejecutar<= 250)
                        {
                                deb_ejecutar++;
                        }
                        else {
                        if(deb_ejecutar>=35)
                        {
                                //Hacer algo
                                ejecutar=1;
                                cont_pasos=cont_pasos*2;
                        }
                        deb_ejecutar=0;
                        }
                }


        } else if(ejecutar == 1)
        {
                if(cont_pasos > 0)
                {
                        if(Procesos[0].TIMER >= Procesos[0].DELTA)
                        {
                                Procesos[0].TIMER = 0;
                                cont_pasos--;
                                Puerto_motor ^= _BV(PULSE);
                        }
                } else
                {
                        Puerto_motor &= ~_BV(PULSE);
                        ejecutar = 0;
                }
        }


    }
}
```