

GRADO EN INGENIERÍA ELÉCTRICA
TRABAJO FIN DE GRADO

***DISEÑO ELÉCTRICO Y CONTROL
ELECTRÓNICO DE MICROCENTRAL
HIDROELÉCTRICA AISLADA***

***DOCUMENTO 6 – ANEXO IV: PROGRAMAS DE CONTROL
ELC CON ARDUINO***

Alumno: Rozas Holgado, Iñigo

Director: Sainz de Murieta Mangado, Joseba

Curso: 2017-2018

Fecha: 28/06/2018

ÍNDICE

| | |
|---|----|
| 1 Programa de control monofásico..... | 1 |
| 2 Programa de control trifásico con carga por fase..... | 5 |
| 3 Programa de control trifásico con única carga..... | 10 |

1 Programa de control monofásico

```
//***** ADICIÓN DE LIBRERÍAS PARA LCD *****  
  
#include <LiquidCrystal.h>  
  
//***** DEFINICIÓN DE PINES *****  
  
#define rs 8  
#define en 9  
#define D4 4  
#define D5 5  
#define D6 6  
#define D7 7 //Pines de conexión del LCD  
#define frec 47 //Pin de entrada para la lectura de la frecuencia  
#define PWM_Pin 2 //Pin de salida del PWM  
#define V A1 //Pin de entrada para la lectura de Vr  
#define I A4 //Pin de entrada para la lectura de I1  
LiquidCrystal lcd(rs, en, D4, D5, D6, D7); //Inicialización de la librería para el LCD  
  
//***** DEFINICIÓN DE VARIABLES *****  
  
double AcsOffset=2.27; //Offset del sensor de corriente  
double Sensibl=0.088; //Sensibilidad del sensor de corriente  
double I1; //Valor de la corriente  
double In=21.8; //Valor de la corriente nominal  
volatile unsigned int cont_t=0; //Contador para el calculo de la frecuencia  
volatile unsigned int f; //Valor de la frecuencia medida  
double fn = 50; //Valor de la frecuencia nominal  
double kf; //Control P de la frecuencia  
int V1; //Valor de la tensión  
double W1; //Valor de la potencia activa  
double k1; //Valor del control de la corriente
```

```
//***** INICIO DEL SETUP *****  
  
void setup() {  
  
//***** DEFINICION DE PINES COMO IN/OUT *****  
  
pinMode (D4, OUTPUT); pinMode (D5, OUTPUT);pinMode (D6, OUTPUT);pinMode (D7,  
OUTPUT);  
  
pinMode (rs, OUTPUT);pinMode (en, OUTPUT);  
  
pinMode (V, INPUT);  
  
pinMode (I, INPUT);  
  
pinMode (PWM_Pin, OUTPUT);  
  
pinMode (frec, INPUT);  
  
  
//***** CONFIGURACION DEL TIMER 4 COMO FUENTE DE INTERRUPCION  
*****  
  
TCCR4A=B00000000; //WGM41=0 y WGM40=0  
TCCR4B=B00001001; //001-1 010-8 011-64 100-256 101-1024  
OCR4A=16000-1; //T=(1/Fosc)*2*PS*(1+OCR4A) ----> CONFIGURADO A 2ms  
TIMSK4=B00000010;  
  
  
//***** CONFIGURACION DEL TIMER 5 COMO CONTADOR DE FRECUENCIA (pin 47)  
*****  
  
TCCR5A = B00000000; //Modo Normal sin pines de salida  
TCCR5B = B01000111; //Modo contador con flanco de subida en T5  
TCNT5 = 0;  
  
  
//***** CONFIGURACION DEL TIMER 3 COMO GENERADOR DE PWM (pin 4)  
*****  
  
TCCR3A = B00100010; //phase and frequency correction pwm  
TCCR3B = B00010010; //PS=8  
ICR3 = 200; //Periodo de la PWM en us ----> CONFIGURADO A 5000Hz  
OCR3B = 0; //Duty Cycle inicial
```

```
//***** MENSAJE INICIAL DE LA LCD *****  
lcd.begin(16,2);  
lcd.setCursor(0,0);lcd.print("V1:");  
lcd.setCursor(6,0);lcd.print("V");  
lcd.setCursor(8,0);lcd.print("I1:");  
lcd.setCursor(15,0);lcd.print("A");  
lcd.setCursor(0,1);lcd.print("P:");  
lcd.setCursor(5,1);lcd.print("kW");  
lcd.setCursor(8,1);lcd.print("f:");  
lcd.setCursor(14,1);lcd.print("Hz");  
}  
  
//***** INICIO DEL LOOP PRINCIPAL *****  
void loop() {  
  
//***** ESCRITURA DE VARIABLES EN LCD *****  
lcd.setCursor(3,0);lcd.print(" ");  
lcd.setCursor(3,0);lcd.print(V1);  
lcd.setCursor(11,0);lcd.print(" ");  
lcd.setCursor(11,0);lcd.print(I1,1);  
lcd.setCursor(2,1);lcd.print(" ");  
lcd.setCursor(2,1);lcd.print(W1,1);  
lcd.setCursor(10,1);lcd.print(" ");  
lcd.setCursor(10,1);lcd.print(f);  
delay(200);  
  
}
```

```
//***** VECTOR DE INTERRUPCIÓN CONTROLADO POR  
TIMER4*****
```

```
ISR(TIMER4_COMPA_vect){
```

```
cont_t++; //Cuenta 1 cada vez que se activa la interrupción
```

```
if (cont_t == 1000){ //Comprobación de cont_t
```

```
f = TCNT5; //Cálculo de la frecuencia
```

```
kf=(fn/f); //Cálculo de kf
```

```
TCNT5 = 0;
```

```
cont_t = 0;
```

```
}
```

```
V1 = map(analogRead(V),0,1023,0,287.5); //Mapeo de la tensión de digital a analogico
```

```
double Va =(analogRead(I)*5.0/1023); //Mapeo de la tensión del sensor de intensidad
```

```
I1 = (Va-AcsOffset)/Sensibl; //Conversión de tensión a intensidad
```

```
W1 =(I1*V1)/1000; //Cálculo de la potencia
```

```
k1=(((I1/In)))*200; //Cálculo de k1
```

```
OCR3B=k1*kf; //Cálculo del duty cycle
```

```
OCR3B=constrain(OCR3B,0,200); //Limitación del duty cycle entre 0 y 200
```

```
}
```



2 Programa de control trifásico con carga por fase

```
//***** ADICIÓN DE LIBRERÍAS PARA LCD *****
```

```
#include <LiquidCrystal.h>
```

```
//***** DEFINICIÓN DE PINES *****
```

```
#define rs 10
```

```
#define en 11
```

```
#define D4 6
```

```
#define D5 7
```

```
#define D6 8
```

```
#define D7 9 //Pines de conexión del LCD
```

```
#define freq 47 //Pin de entrada para la lectura de la frecuencia
```

```
#define PWM_Pin_A 5 //Pin de salida del PWM
```

```
#define PWM_Pin_B 2 //Pin de salida del PWM
```

```
#define PWM_Pin_C 3 //Pin de salida del PWM
```

```
#define Vr A1 //Pin de entrada para la lectura de Vr
```

```
#define Vs A2 //Pin de entrada para la lectura de Vs
```

```
#define Vt A3 //Pin de entrada para la lectura de Vt
```

```
#define Ir A4 //Pin de entrada para la lectura de I1
```

```
#define Is A5 //Pin de entrada para la lectura de I2
```

```
#define It A6 //Pin de entrada para la lectura de I3
```

```
LiquidCrystal lcd(rs, en, D4, D5, D6, D7);
```

```
//***** DEFINICIÓN DE VARIABLES *****
```

```
double AcsOffset=2.27; //Offset del sensor de corriente
```

```
double Sensibl=0.129; //Sensibilidad del sensor de corriente
```

```
double I1, I2, I3; //Valor de las corrientes de cada fase
```

```
double In=7.2; //Valor de la corriente nominal
```

```
volatile unsigned int cont_t=0; //Contador para el calculo de la frecuencia
```

```
volatile unsigned int f; //Valor de la frecuencia medida
```

```
double fn = 50; //Valor de la frecuencia nominal
```

```
double kf; //Control P de la frecuencia
```

```
int V1, V2, V3;          //Valor de las tensiones de cada fase
double W1, W2, W3, W;    //Valor de la potencia activa de cada fase
double k1, k2, k3;      //Valor del control de la corriente de cada fase

//***** INICIO DEL SETUP *****

void setup() {

//***** DEFINICION DE PINES COMO IN/OUT *****

pinMode (D4, OUTPUT); pinMode (D5, OUTPUT);pinMode (D6, OUTPUT);pinMode (D7,
OUTPUT);
pinMode (rs, OUTPUT);pinMode (en, OUTPUT);
pinMode (Vr, INPUT); pinMode (Vs, INPUT); pinMode (Vt, INPUT);
pinMode (Ir, INPUT); pinMode (Is, INPUT); pinMode (It, INPUT);
pinMode (PWM_Pin_A, OUTPUT);pinMode (PWM_Pin_B, OUTPUT);pinMode (PWM_Pin_C,
OUTPUT);
pinMode (frec, INPUT);

//***** CONFIGURACION DEL TIMER 4 COMO FUENTE DE INTERRUPCION
*****

TCCR4A=B00000000; //WGM41=0 y WGM40=0
TCCR4B=B00001001; //001-1 010-8 011-64 100-256 101-1024
OCR4A=32000-1; //T=(1/Fosc)*2*PS*(1+OCR4A) ----> CONFIGURADO A 4ms
TIMSK4=B00000010;

//***** CONFIGURACION DEL TIMER 5 COMO CONTADOR DE FRECUENCIA (pin 4)
*****

TCCR5A = B00000000; //Modo Normal sin pines de salida
TCCR5B = B01000111; //Modo contador con flanco de subida en T5
TCNT5 = 0;

//***** CONFIGURACION DEL TIMER 3 COMO GENERADOR DE PWM (pin 4)
*****

TCCR3A = B10101010; //phase and frequency correction pwm
TCCR3B = B00010010; //PS=8
```



```
ICR3 = 200; //Periodo de la PWM en us ----> CONFIGURADO A 2ms
OCR3A = 0; //Duty Cycle inicial
OCR3B = 0; //Duty Cycle inicial
OCR3C = 0; //Duty Cycle inicial
```

```
//***** MENSAJE INICIAL DE LA LCD *****
```

```
lcd.begin(16,4);
lcd.setCursor(0,0);lcd.print("V1:");
lcd.setCursor(0,1);lcd.print("V2:");
lcd.setCursor(0,2);lcd.print("V3:");
lcd.setCursor(6,0);lcd.print("V");
lcd.setCursor(6,1);lcd.print("V");
lcd.setCursor(6,2);lcd.print("V");
lcd.setCursor(8,0);lcd.print("I1:");
lcd.setCursor(8,1);lcd.print("I2:");
lcd.setCursor(8,2);lcd.print("I3:");
lcd.setCursor(15,0);lcd.print("A");
lcd.setCursor(15,1);lcd.print("A");
lcd.setCursor(15,2);lcd.print("A");
lcd.setCursor(0,3);lcd.print("P:");
lcd.setCursor(5,3);lcd.print("kW");
lcd.setCursor(8,3);lcd.print("f:");
lcd.setCursor(14,3);lcd.print("Hz");
```

```
}
```

```
//***** INICIO DEL LOOP PRINCIPAL *****
```

```
void loop() {
```

```
//***** ESCRITURA DE VARIABLES EN LCD *****
```

```
lcd.setCursor(3,0);lcd.print(" ");
lcd.setCursor(3,0);lcd.print(V1);
lcd.setCursor(3,1);lcd.print(" ");
```

```
lcd.setCursor(3,1);lcd.print(V2);  
lcd.setCursor(3,2);lcd.print(" ");  
lcd.setCursor(3,2);lcd.print(V3);  
lcd.setCursor(11,0);lcd.print(" ");  
lcd.setCursor(11,0);lcd.print(l1,1);  
lcd.setCursor(11,1);lcd.print(" ");  
lcd.setCursor(11,1);lcd.print(l2,1);  
lcd.setCursor(11,2);lcd.print(" ");  
lcd.setCursor(11,2);lcd.print(l3,1);  
lcd.setCursor(2,3);lcd.print(" ");  
lcd.setCursor(2,3);lcd.print(W,1);  
lcd.setCursor(10,3);lcd.print(" ");  
lcd.setCursor(10,3);lcd.print(f);  
delay(200);  
}
```

```
//***** VECTOR DE INTERRUPCIÓN CONTROLADO POR  
TIMER4*****  
ISR(TIMER4_COMPA_vect){  
cont_t++; //Cuenta 1 cada vez que se activa la interrupción  
if (cont_t == 500){ //Comprobación de cont_t  
f = TCNT5; //Cálculo de la frecuencia  
kf=(fn/f); //Cálculo de kf  
TCNT5 = 0;  
cont_t = 0;  
}  
V1 = map(analogRead(Vr),0,1023,0,287.5);  
V2 = map(analogRead(Vs),0,1023,0,287.5);  
V3 = map(analogRead(Vt),0,1023,0,287.5); //Mapeo de la tensión  
double Va =(analogRead(Ir)*5.0/1023);  
I1 = (Va-AcsOffset)/Sensibl;  
double Vb =(analogRead(Is)*5.0/1023);  
I2 = (Vb-AcsOffset)/Sensibl;
```

```
double Vc =(analogRead(It)*5.0/1023); //Mapeo de la tensión del sensor de intensidad
I3 = (Vc-AcsOffset)/Sensibl; //Conversión de tensión a intensidad
W1 =(I1*V1)/1000;
W2 =(I2*V2)/1000;
W3 =(I3*V3)/1000; //Cálculo de la potencia por fase
W = W1+W2+W3; //Cálculo de la potencia total
k1=(((I1/In)))2*200; //Cálculo de k1
k2=(((I2/In)))2*200; //Cálculo de k2
k3=(((I3/In)))2*200; //Cálculo de k3
OCR3A=k1*kf; //Cálculo del duty cycle 1
OCR3A=constrain(OCR3A,0,200); //Limitación del duty cycle entre 0 y 200
OCR3B=k2*kf; //Cálculo del duty cycle 2
OCR3B=constrain(OCR3B,0,200); //Limitación del duty cycle entre 0 y 200
OCR3C=k3*kf; //Cálculo del duty cycle 3
OCR3C=constrain(OCR3C,0,200); //Limitación del duty cycle entre 0 y 200
}
```

3 Programa de control trifásico con única carga

```
//***** ADICIÓN DE LIBRERÍAS PARA LCD *****
```

```
#include <LiquidCrystal.h>
```

```
//***** DEFINICIÓN DE PINES *****
```

```
#define rs 10
```

```
#define en 11
```

```
#define D4 6
```

```
#define D5 7
```

```
#define D6 8
```

```
#define D7 9 //Pines de conexión del LCD
```

```
#define frec 47 //Pin de entrada para la lectura de la frecuencia
```

```
#define PWM_Pin 5 //Pin de salida del PWM
```

```
#define Vr A1 //Pin de entrada para la lectura de Vr
```

```
#define Vs A2 //Pin de entrada para la lectura de Vs
```

```
#define Vt A3 //Pin de entrada para la lectura de Vt
```

```
#define Ir A4 //Pin de entrada para la lectura de I1
```

```
#define Is A5 //Pin de entrada para la lectura de I2
```

```
#define It A6 //Pin de entrada para la lectura de I3
```

```
LiquidCrystal lcd(rs, en, D4, D5, D6, D7); //Inicialización de la librería para el LCD
```

```
//***** DEFINICIÓN DE VARIABLES *****
```

```
double AcsOffset=2.27; //Offset del sensor de corriente
```

```
double Sensibl=0.129; //Sensibilidad del sensor de corriente
```

```
double I1, I2, I3; //Valor de las corrientes de cada fase
```

```
double In=7.2; //Valor de la corriente nominal
```

```
volatile unsigned int cont_t=0; //Contador para el calculo de la frecuencia
```

```
volatile double f; //Valor de la frecuencia medida
```

```
int fn = 50; //Valor de la frecuencia nominal
```

```
double kf; //Control P de la frecuencia
```

```
int V1, V2, V3; //Valor de las tensiones de cada fase
```

```
int V1m, V2m, V3m; //Valor máximo de las tensiones de cada fase
```

double W1, W2, W3, W; //Valor de la potencia activa de cada fase
double k1, k2, k3; //Valor del control de la corriente de cada fase

```
//***** INICIO DEL SETUP *****
```

```
void setup() {
```

```
//***** DEFINICION DE PINES COMO IN/OUT *****
```

```
pinMode (D4, OUTPUT); pinMode (D5, OUTPUT);pinMode (D6, OUTPUT);pinMode (D7,  
OUTPUT);
```

```
pinMode (rs, OUTPUT);pinMode (en, OUTPUT);
```

```
pinMode (Vr, INPUT); pinMode (Vs, INPUT); pinMode (Vt, INPUT);
```

```
pinMode (Ir, INPUT); pinMode (Is, INPUT); pinMode (It, INPUT);
```

```
pinMode (PWM_Pin, OUTPUT);
```

```
pinMode (frec, INPUT);
```

```
//***** CONFIGURACION DEL TIMER 4 COMO FUENTE DE INTERRUPCION  
*****
```

```
TCCR4A=B00000000; //WGM41=0 y WGM40=0
```

```
TCCR4B=B00001001; //001-1 010-8 011-64 100-256 101-1024
```

```
OCR4A=63999; //T=(1/Fosc)*2*PS*(1+OCR4A) -----> CONFIGURADO A 8ms
```

```
TIMSK4=B00000010;
```

```
//***** CONFIGURACION DEL TIMER 5 COMO CONTADOR DE FRECUENCIA (pin 47)  
*****
```

```
TCCR5A = B00000000; //Modo Normal sin pines de salida
```

```
TCCR5B = B01000111; //Modo contador con flanco de subida en T5
```

```
TCNT5 = 0;
```

```
//***** CONFIGURACION DEL TIMER 3 COMO GENERADOR DE PWM (pin 4)  
*****
```

```
TCCR3A = B10101010; //phase and frequency correction pwm
```

```
TCCR3B = B00010010; //PS=8
```

```
ICR3 = 200; //Periodo de la PWM en us ----> CONFIGURADO A 2ms
```

```
OCR3A = 0; //Duty Cycle inicial
```

```
//***** MENSAJE INICIAL DE LA LCD *****
```

```
lcd.begin(16,4);  
lcd.setCursor(0,0);lcd.print("V1:");  
lcd.setCursor(0,1);lcd.print("V2:");  
lcd.setCursor(0,2);lcd.print("V3:");  
lcd.setCursor(6,0);lcd.print("V");  
lcd.setCursor(6,1);lcd.print("V");  
lcd.setCursor(6,2);lcd.print("V");  
lcd.setCursor(8,0);lcd.print("I1:");  
lcd.setCursor(8,1);lcd.print("I2:");  
lcd.setCursor(8,2);lcd.print("I3:");  
lcd.setCursor(15,0);lcd.print("A");  
lcd.setCursor(15,1);lcd.print("A");  
lcd.setCursor(15,2);lcd.print("A");  
lcd.setCursor(0,3);lcd.print("P:");  
lcd.setCursor(5,3);lcd.print("kW");  
lcd.setCursor(8,3);lcd.print("f:");  
lcd.setCursor(14,3);lcd.print("Hz");  
}
```

```
//***** INICIO DEL LOOP PRINCIPAL *****
```

```
void loop() {
```

```
//***** ESCRITURA DE VARIABLES EN LCD *****
```

```
lcd.setCursor(3,0);lcd.print(" ");  
lcd.setCursor(3,0);lcd.print(V1m);  
lcd.setCursor(3,1);lcd.print(" ");  
lcd.setCursor(3,1);lcd.print(V2m);  
lcd.setCursor(3,2);lcd.print(" ");  
lcd.setCursor(3,2);lcd.print(V3m);
```

```
lcd.setCursor(11,0);lcd.print(" ");  
lcd.setCursor(11,0);lcd.print(11,1);  
lcd.setCursor(11,1);lcd.print(" ");  
lcd.setCursor(11,1);lcd.print(12,1);  
lcd.setCursor(11,2);lcd.print(" ");  
lcd.setCursor(11,2);lcd.print(13,1);  
lcd.setCursor(2,3);lcd.print(" ");  
lcd.setCursor(2,3);lcd.print(W,1);  
lcd.setCursor(10,3);lcd.print(" ");  
lcd.setCursor(10,3);lcd.print(f,0);  
delay(100);
```

```
}
```

```
/******* VECTOR DE INTERRUPCIÓN CONTROLADO POR  
TIMER4*****
```

```
ISR(TIMER4_COMPA_vect){
```

```
cont_t++; //Cuenta 1 cada vez que se activa la interrupción  
if (cont_t == 250){ //Cuenta 1 cada vez que se activa la interrupción  
f = TCNT5; //Cálculo de la frecuencia  
kf=(f/fn); //Cálculo de kf  
TCNT5 = 0;  
cont_t = 0;  
}
```

```
V1 = map(analogRead(Vr),0,1023,0,287.5);
```

```
V1m = max(V1,230);
```

```
V2 = map(analogRead(Vs),0,1023,0,287.5);
```

```
V2m = max(V2,230);
```

```
V3 = map(analogRead(Vt),0,1023,0,287.5); //Mapeo de la tensión
```

```
V3m = max(V3,230); //Variable para mostrar tensión máxima siempre
```

```
double Va =(analogRead(Ir)*5.0/1023);
```

```
I1 = (Va-AcsOffset)/Sensibl;
```

```

double Vb =(analogRead(Is)*5.0/1023);
I2 = (Vb-AcsOffset)/Sensibl;
double Vc =(analogRead(It)*5.0/1023); //Mapeo de la tensión del sensor de intensidad
I3 = (Vc-AcsOffset)/Sensibl; //Conversión de tensión a intensidad
W1 =(I1*V1m)/1000;
W2 =(I2*V2m)/1000;
W3 =(I3*V3m)/1000; //Cálculo de la potencia por fase
W = W1+W2+W3; //Cálculo de la potencia total
k1=((1-(I1/In)))^200; //Cálculo de k1
k2=((1-(I2/In)))^200; //Cálculo de k2
k3=((1-(I3/In)))^200; //Cálculo de k3

//***** CONTROL DEL PWM POR FASE *****
if ((V1 > V2) && (V3 <= V1)) //Condiciones fase R
{
OCR3A = k1*kf; //Cálculo del k1 e introducción en OCR3A
OCR3A = constrain(OCR3A,0,200); //Limitación del duty cycle entre 0 y 200
}
if ((V2 > V3) && (V1 <= V2)) //Condiciones fase S
{
OCR3A= k2*kf; //Limitación del duty cycle entre 0 y 200
OCR3A =constrain(OCR3A,0,200); //Cálculo de k2 introducción en OCR3A
}
if ((V3 > V1) && (V2 <= V3)) //CONDICIONES fase T
{
OCR3A= k3*kf; //Limitación del duty cycle entre 0 y 200
OCR3A = constrain(OCR3A,0,200); //Limitación del duty cycle entre 0 y 200
}
}

```