

Práctica de control con la tarjeta
STM32F411E Discovery:

Caracterización y control PID de un
motor de corriente continua

Realizado por: Pablo Buestán Andrade.

Fecha: febrero de 2018

Índice de Contenidos

Introducción.....	1
Desarrollo.....	1
HARDWARE	1
1. Diagrama de bloques.....	1
2. Señal generada de tipo PWM.....	2
3. Etapa de potencia, puente en H y sentido de giro.....	3
4. Medición de velocidad	4
SOFTWARE	6
1. Entorno de desarrollo “System Workbench for STM32”	6
1.1. Caracterización de un motor de CC.	6
1.2. Control PID de un motor de CC.....	9
2. Filtrado de datos adquiridos	11
Conclusiones	14
Recomendaciones.....	15
Bibliografía.....	17
Anexos	18

Índice de Figuras

Figura 1: Diagrama de bloques	1
Figura 2: Señal PWM.....	2
Figura 3: Circuito etapa de potencia, puente en H y sentido de giro.....	3
Figura 4: Esquema mecánico del motor (Profesores del Dpto. de Ingeniería Electrónica de la ETSI Telecomunicación de la UPM, 2017).	4
Figura 5: Circuito optoacoplador (Profesores del Dpto. de Ingeniería Electrónica de la ETSI Telecomunicación de la UPM, 2017).	4
Figura 6: Circuito para la medición de velocidad.....	5
Figura 7: Programa principal de caracterización y control PID de un motor CC .	6
Figura 8: Configuración de PuTTY	7
Figura 9: Menú principal	8
Figura 10: Menú prueba motor	8
Figura 11: Menú caracterización	8
Figura 12: Curva de caracterización del motor a un ciclo de 85/255.....	9
Figura 13: Menú control PID	10
Figura 14: Respuesta del motor a una velocidad de 60 rev/seg	11
Figura 15: Menú cambiar parámetros PID.....	11
Figura 16: Muestras almacenadas.....	12
Figura 17: Guardar sesión PuTTY	12
Figura 18: Carpeta Pruebas_MOTOR.....	13
Figura 19: Carpeta Pruebas_MOTOR con ficheros de pruebas	13
Figura 20: Conexión correcta del periférico USB a máquina virtual	15
Figura 21: Error de conexión con la tarjeta STM32F411E Discovery	15
Figura 22: Propiedades del sistema de Windows 7	16
Figura 23: Administrador de dispositivos de Windows 7	16

Introducción

A lo largo de esta práctica se propone al estudiante aplicar los conocimientos adquiridos en la parte teórica de la asignatura “Control de procesos” para:

- Realizar la caracterización de un motor de CC¹, calculando su curva velocidad/tensión y la respuesta al escalón.
- Controlar la velocidad de giro de un motor de CC, usando un controlador digital de tipo PID.

Para ello, se **deberá** realizar diferentes pruebas en un circuito que consta de un generador de señal, etapa de potencia capaz de hacer girar el motor a diferentes velocidades y un detector óptico que realiza la medición de las velocidades de giro.

La tensión de entrada será generada por la tarjeta STM32F411E Discovery, basada en un microcontrolador STM32F411VET6 y un programa diseñado a tal efecto. Para poder ejecutar este programa, el alumno utilizará el IDE² “System Workbench for STM32”.

Desarrollo

A continuación, se explica el funcionamiento tanto del hardware, como del software, de manera que el alumno este familiarizado con los instrumentos a utilizar.

HARDWARE

1. Diagrama de bloques

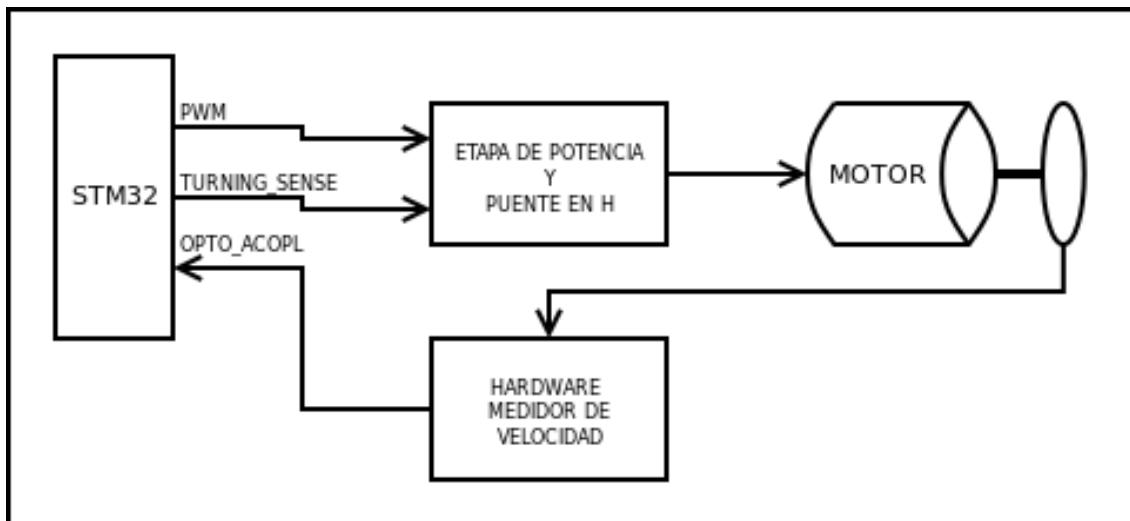


Figura 1: Diagrama de bloques

En la figura anterior se observa que la tarjeta STM32F411E Discovery es la encargada de generar la tensión de entrada, que, a través de la etapa de potencia y puente en H,

¹ CC: Corriente continua

² IDE: Entorno de desarrollo.

el motor tenga la capacidad de girar; y a su vez, la señal adquirida por el hardware de medición de velocidad regrese al sistema empotrado.

2. Señal generada de tipo PWM

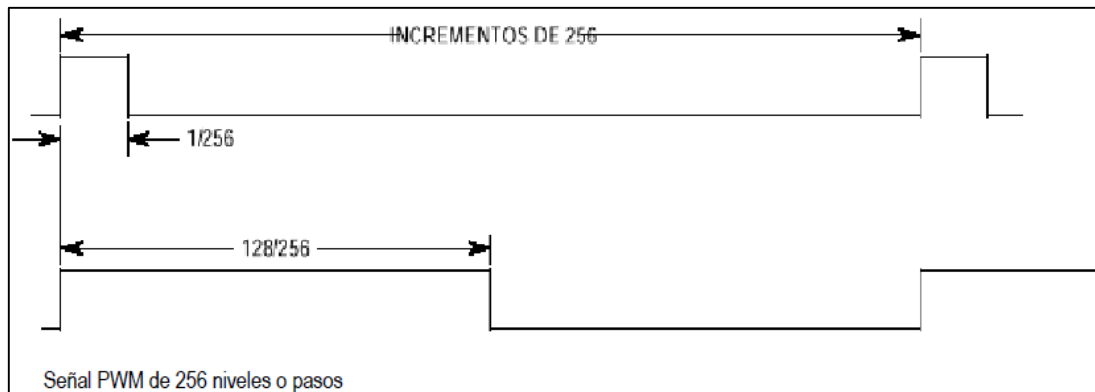


Figura 2: Señal PWM

La señal generada es de tipo PWM³. Dado que el motor es un sistema mecánico relativamente lento, si se genera una señal periódica cuadrada (figura 2), de una frecuencia suficientemente elevada (1KHz, por ejemplo) el motor responderá a la tensión o corriente continua promedio que se le aplique. El valor medio de una señal PWM es directamente proporcional a su ciclo de trabajo (Profesores del Dpto. de Ingeniería Electrónica de la ETSI Telecomunicación de la UPM, 2017). El ciclo de trabajo es la relación entre el tiempo que está en nivel alto dentro de un periodo, dividido por la duración de dicho periodo.

Entre los periféricos de cualquier microcontrolador destaca la presencia de temporizadores de propósito general (TIMER). En este caso, los registros de gestión de timer están en el módulo TIM. En la tarjeta STM32F411E Discovery se tiene 4 timer (TIM 2, 3, 4 y 5).

Se pueden utilizar como: generadores de interrupciones temporales (cuando terminan la cuenta), medida de ancho de pulsos externos (input compare) o generadores de señales (output compare o PWM) (Malagón Marzo & Fernández, 2017). Para este sistema se ha utilizado los timers 3 y 4 configurados como input compare y PWM respectivamente. Los valores generados están compuestos en un conjunto discreto de ciclos de trabajo (0, 1/255, 2/255, 3/255... 255/255), con los que el alumno realizará pruebas de velocidad cuando se aplique cierta tensión/corriente (linealmente proporcional al ciclo de trabajo).

³ PWM: Modulación por ancho de pulso.

3. Etapa de potencia, puente en H y sentido de giro

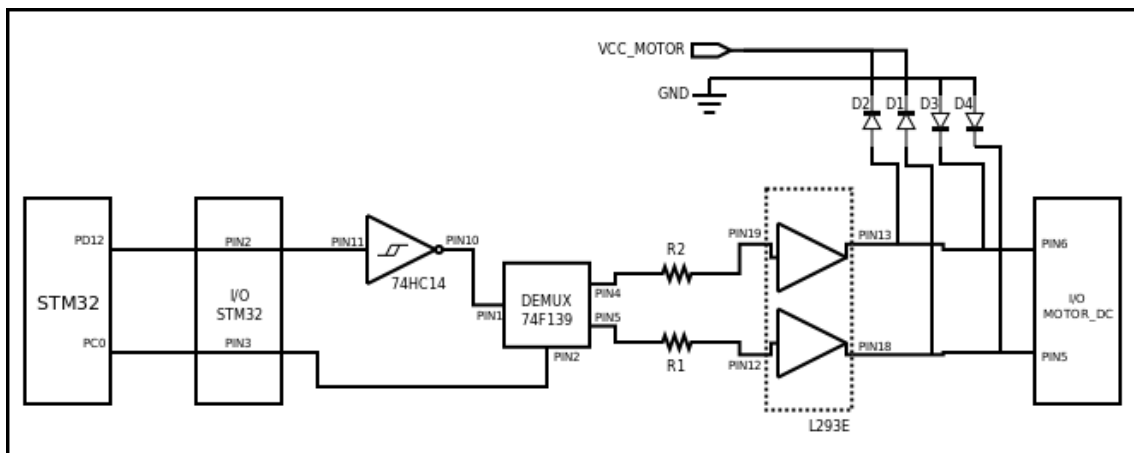


Figura 3: Circuito etapa de potencia, puente en H y sentido de giro.

Analizando la figura anterior:

- La tarjeta STM32F411E Discovery por la GPIO PD12 genera la señal PWM, mientras que por la GPIO PC0 escoge el sentido de giro del motor, habilitando (1 lógico) o deshabilitando (0 lógico) este puerto.
- PD12 se conecta con PIN2, y PC0 con PIN3 de las ranuras de conexión I/O STM32F4.
- La señal PWM debe de ser lo más “limpia” posible; es decir, **mientras** menos ruido se añade, será mejor. Para reducir el ruido se ha implementado un inversor Schmitt-trigger (74HC14). Este CI⁴ genera una señal conformada digitalmente (0 o 5 voltios).
- La señal obtenida luego del CI 74HC14 se conecta a la entrada del CI 74F139 (PIN1); un demultiplexor que enviará la señal PWM por un puerto u otro (PIN4 o PIN5 respectivamente), dependiendo del valor que este en PIN2.
- La salida PWM no es capaz de proporcionar suficiente potencia para mover el motor, por lo cual es necesario un driver que proporciona la corriente adecuada al motor; además de controlar el sentido de giro. Este driver es el CI L293E, cuyas entradas son PIN19 y PIN12. Estos pines se conectan con las salidas del demultiplexor mediante una resistencia de valor pequeño R1 y R2 (220 Ω cada una).
- Por último, para proteger al motor se ha realizado un circuito de diodos (cuatro diodos 1N4004) que se conectan entre las salidas PIN13 y PIN18 del CI L293E y las ranuras de conexión I/O MOTOR/DC PIN6 y PIN5.

⁴ CI: Circuito integrado.

4. Medición de velocidad

Para medir la velocidad, se debe generar una señal, cuya frecuencia debe ser proporcional a dicha velocidad.

El motor lleva acoplado un disco con un orificio como se muestra en la figura 4. Al girar el motor, el agujero pasa **un** por una pieza en U. Este aparato es el sistema optoacoplador que consta de un fotoemisor y un fotodetector.

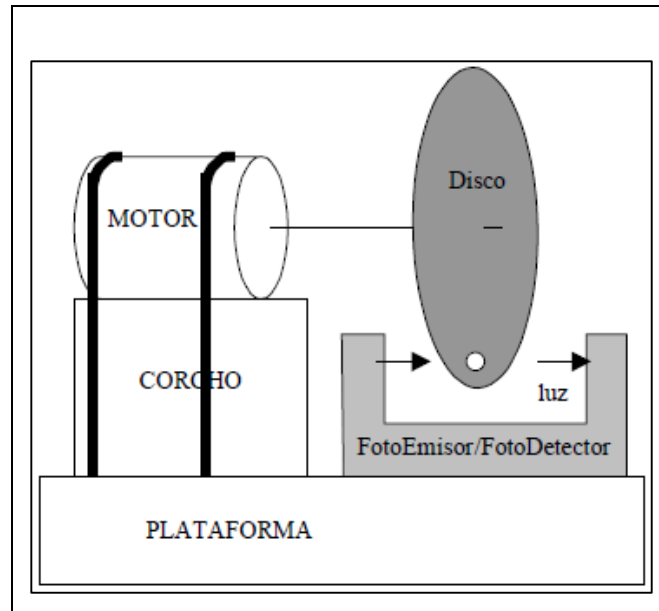


Figura 4: Esquema mecánico del motor (Profesores del Dpto. de Ingeniería Electrónica de la ETSI Telecomunicación de la UPM, 2017).

La luz enviada por el fotoemisor atraviesa (o no) el orificio, cambiando su estado eléctrico (Profesores del Dpto. de Ingeniería Electrónica de la ETSI Telecomunicación de la UPM, 2017).

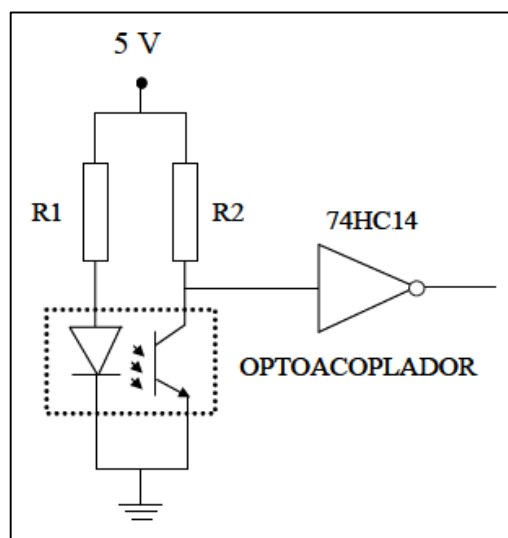


Figura 5: Circuito optoacoplador (Profesores del Dpto. de Ingeniería Electrónica de la ETSI Telecomunicación de la UPM, 2017).

En la figura 5 se observa que el diodo emisor debe estar conectado a una resistencia R1; esta no debe de ser muy grande (200 Ω) para que emita luz con intensidad y sea fácilmente leído por el detector. A través de la resistencia R2 (500 Ω) se consigue que la señal este en nivel bajo cuando no reciba luz y viceversa.

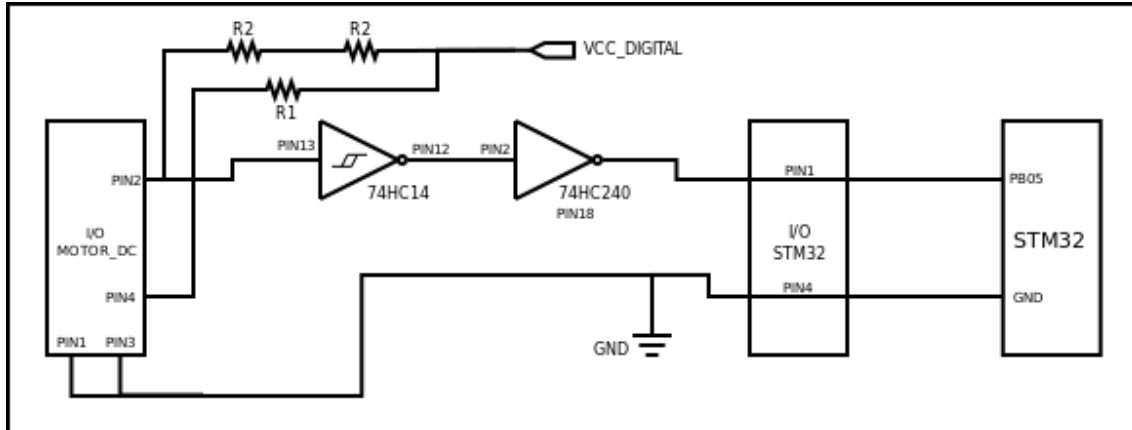


Figura 6: Circuito para la medición de velocidad

De la figura anterior:

- R1 (200 Ω) está conectado al PIN4, y R2 (dos resistencias de 1k Ω) a PIN2 de las ranuras de conexión I/O MOTOR_DC.
- A fin de minimizar el ruido que afecta negativamente la estimación de la velocidad se procede a utilizar nuevamente un inversor Schmitt-trigger (74HC14).
- La salida del CI 74HC14 es la generadora de los pulsos que se conecta a la tarjeta STM32F411E Discovery. A fin de proteger el microprocesador se interpondrá un buffer (74HC240).
- La salida del buffer se conecta con el PIN1 de la ranura de conexión I/O STM32 que está conectada a la GPIO PB05 de la STM32F411E Discovery.
- Se cortocircuita todos los GND.

Adicional, para minimizar aún más el ruido:

- Se colocan condensadores de filtrado de alimentación entre V_{CC} y GND.
- Se conectan todas las entradas digitales no usadas en los CI.
- Se alimentan con fuentes diferentes el circuito digital y el circuito del motor.
- Se colocan condensadores de 100 o 10nF entre V_{CC} y GND de los CI (Profesores del Dpto. de Ingeniería Electrónica de la ETSI Telecomunicación de la UPM, 2017).

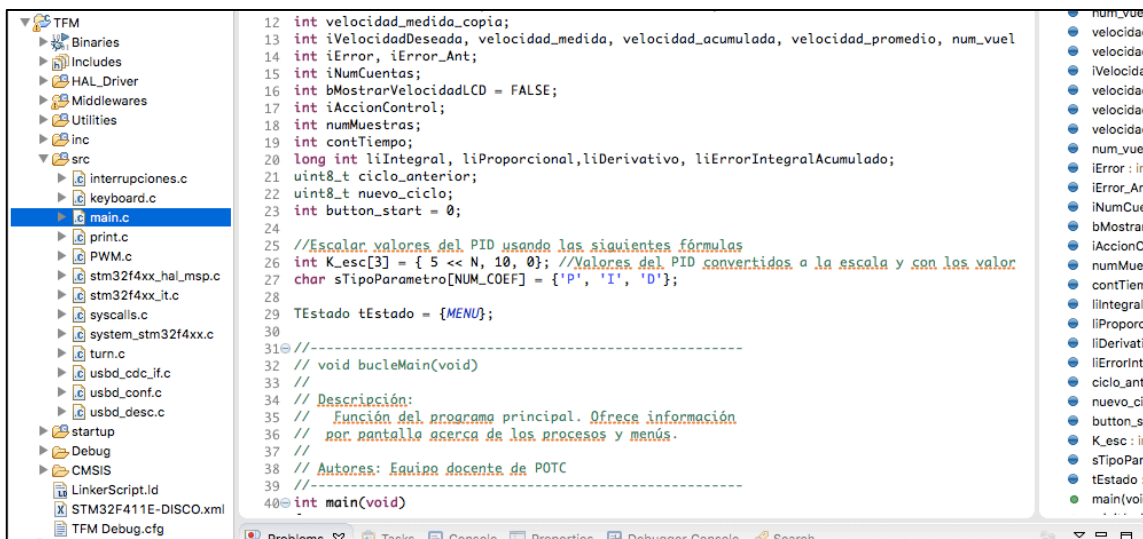
SOFTWARE

1. Entorno de desarrollo "System Workbench for STM32"

1.1. Caracterización de un motor de CC.

En el puesto de trabajo se ha instalado "System Workbench for STM32" y PuTTY (cliente SSH); este último necesario para realizar la comunicación entre la tarjeta STM32F411E Discovery y la PC.

Para cargar correctamente el programa en la tarjeta STM32F411E Discovery, seguir los siguientes pasos:



```
12 int velocidad_medida_copia;
13 int iVelocidadDeseada, velocidad_medida, velocidad_acumulada, velocidad_promedio, num_vuel
14 int iError, iError_Ant;
15 int iNumCuentas;
16 int bMostrarVelocidadLCD = FALSE;
17 int iAccionControl;
18 int numMuestras;
19 int contTiempo;
20 long int liIntegral, liProporcional, liDerivativo, liErrorIntegralAcumulado;
21 uint8_t ciclo_anterior;
22 uint8_t nuevo_ciclo;
23 int button_start = 0;
24
25 //Escalar valores del PID usando las siguientes fórmulas
26 int K_esc[3] = { 5 << N, 10, 0}; //Valores del PID convertidos a la escala y con los valor
27 char sTipoParametro[NUM_COEF] = {'P', 'I', 'D'};
28
29 TEstado tEstado = {MENU};
30
31 //-----
32 // void bucleMain(void)
33 //
34 // Descripción:
35 // Función del programa principal. Ofrece información
36 // por pantalla acerca de los procesos y menús.
37 //
38 // Autores: Equipo docente de POTC
39 //-----
40 int main(void)
```

Figura 7: Programa principal de caracterización y control PID de un motor CC


1. El usuario debe ir a la dirección mostrada en la figura anterior y abrir el fichero main.c.
2. Compilar para comprobar errores.
 - El alumno obtendrá **TRES** errores que deberá solucionar:

<p>Configuración adecuada del PWM:</p> <p>En el fichero PWM.c, tener en cuenta:</p> <ul style="list-style-type: none">• htim4.Init.Period.- Es el periodo de la cuenta. Si la cuenta llega a ese número, se lanza la interrupción y se pone la cuenta con un valor = 0.• htim4.Init.Prescaler.- Se puede configurar para reducir la frecuencia de reloj de entrada del temporizador.• sConfigTim4.Pulse = TIM_CCR1.- Se configura el ciclo de trabajo.
<p>Parámetros PID:</p> <p>En el fichero main.c, en la función "controladorPID" faltan algunas variables marcadas con XXX; se debe de completar correctamente.</p>

- Si se ha configurado correctamente sería posible el compilar y ejecutar el programa adecuadamente, sin embargo, no funcionaría; esto es por un tercer error:

Configuración del reloj:

En el fichero main.c, en la función `__init()` falta “algo”. El alumno debe encontrar el error en esta parte del proyecto.

3. Una vez corregido se compila, se ejecuta y se espera hasta que se cargue correctamente el fichero en la tarjeta STM32F411E Discovery.
4. Dar clic en  para abrir el software PuTTY.

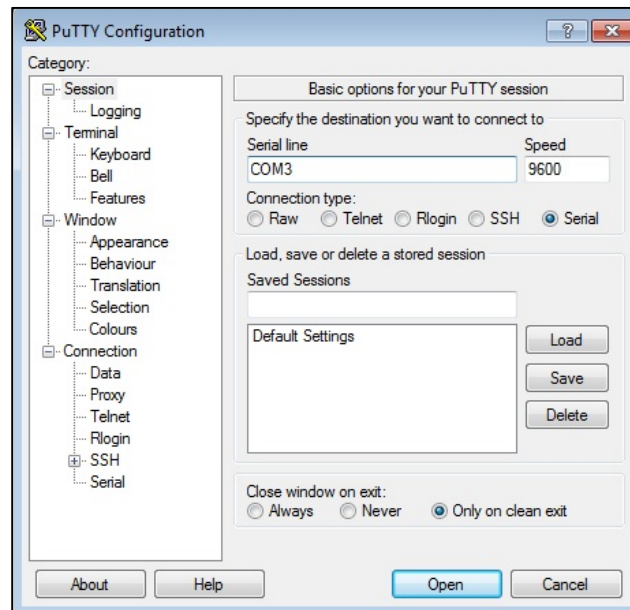
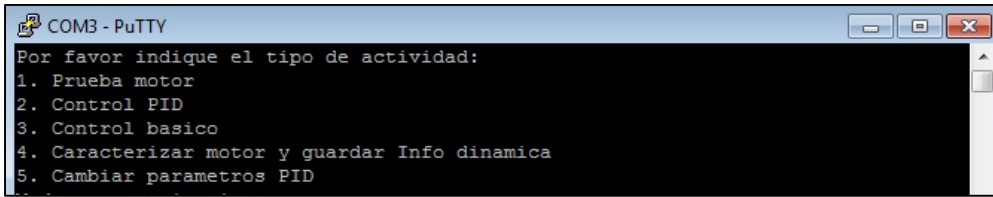


Figura 8: Configuración de PuTTY

5. Una vez abierto, se configura PuTTY, en donde:
 - Serial line: Número de COM configurado (COM3).
 - Speed: 9600.
 - Connection type: Serial.
 - Clic en “open”.

Si todo se ha configurado correctamente, PuTTY abre una pantalla con fondo negro; esto quiere decir que se ha realizado la comunicación entre la PC y la tarjeta STM32F411E Discovery.

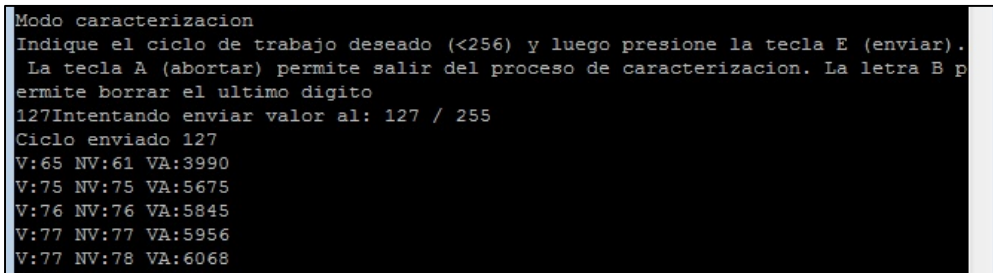
6. Dar clic en el botón azul de la tarjeta STM32F411E Discovery. Se inicializa el proceso y se presenta la siguiente pantalla:



```
COM3 - PuTTY
Por favor indique el tipo de actividad:
1. Prueba motor
2. Control PID
3. Control basico
4. Caracterizar motor y guardar Info dinamica
5. Cambiar parametros PID
```

Figura 9: Menú principal

7. Se puede realizar una prueba del motor tecleando “1”:



```
Modo caracterizacion
Indique el ciclo de trabajo deseado (<256) y luego presione la tecla E (enviar).
La tecla A (abortar) permite salir del proceso de caracterizacion. La letra B p
ermite borrar el ultimo digito
127Intentando enviar valor al: 127 / 255
Ciclo enviado 127
V:65 NV:61 VA:3990
V:75 NV:75 VA:5675
V:76 NV:76 VA:5845
V:77 NV:77 VA:5956
V:77 NV:78 VA:6068
```

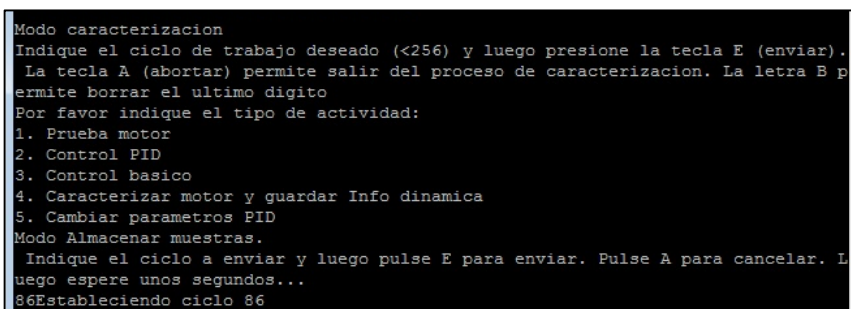
Figura 10: Menú prueba motor

El programa permite incrementar o disminuir el ciclo de trabajo (un número entre 0 y 255). La frecuencia del PWM está programada a 1KHz. Posterior, el usuario presiona la tecla E y el microcontrolador envía el valor de PWM seleccionado y por la pantalla se muestra los datos adquiridos mediante el optoacoplador (en este caso se ha elegido 127/255, un ciclo de trabajo aproximadamente del 50%).

Al momento de enviar el ciclo, las interrupciones son activadas; cada cambio de estado en el optoacoplador se almacena en una variable. La frecuencia del contador de pulsos que fue configurada a 1MHz (TIM3) se divide por la variable mencionada anteriormente para obtener una velocidad medida. Por otra parte, cada vez que esto ocurre el número de vueltas también es almacenado para ser mostrados en pantalla (figura anterior) en donde:

- V: Velocidad medida.
- NV: Número de vueltas.
- VA: Velocidad acumulada.

8. Realizar la caracterización a partir del transitorio (menú 4).



```
Modo caracterizacion
Indique el ciclo de trabajo deseado (<256) y luego presione la tecla E (enviar).
La tecla A (abortar) permite salir del proceso de caracterizacion. La letra B p
ermite borrar el ultimo digito
Por favor indique el tipo de actividad:
1. Prueba motor
2. Control PID
3. Control basico
4. Caracterizar motor y guardar Info dinamica
5. Cambiar parametros PID
Modo Almacenar muestras.
Indique el ciclo a enviar y luego pulse E para enviar. Pulse A para cancelar. L
uego espere unos segundos...
86Estableciendo ciclo 86
```

Figura 11: Menú caracterización

El programa permite caracterizar el motor a partir de un ciclo determinado enviado y esos datos determinados mediante la medición del optoacoplador mostrarlos en pantalla para poder generar una curva de velocidad.

El motor se pondrá en movimiento si se ha aplicado un ciclo adecuado. Una vez que la velocidad se ha estabilizado o han pasado más de 10 segundos, se debe de detener el proceso pulsando la tecla A. Se mostrará las velocidades medidas durante el tiempo de ejecución y se regresará al menú principal, imprimiendo por pantalla además el valor de la velocidad media final y la velocidad mediana. Estos datos pueden ser exportados para luego introducirlos en Microsoft Excel. Al final se puede generar una curva de velocidad donde se puede apreciar la respuesta al escalón (Profesores del Dpto. de Ingeniería Electrónica de la ETSI Telecomunicación de la UPM, 2017).

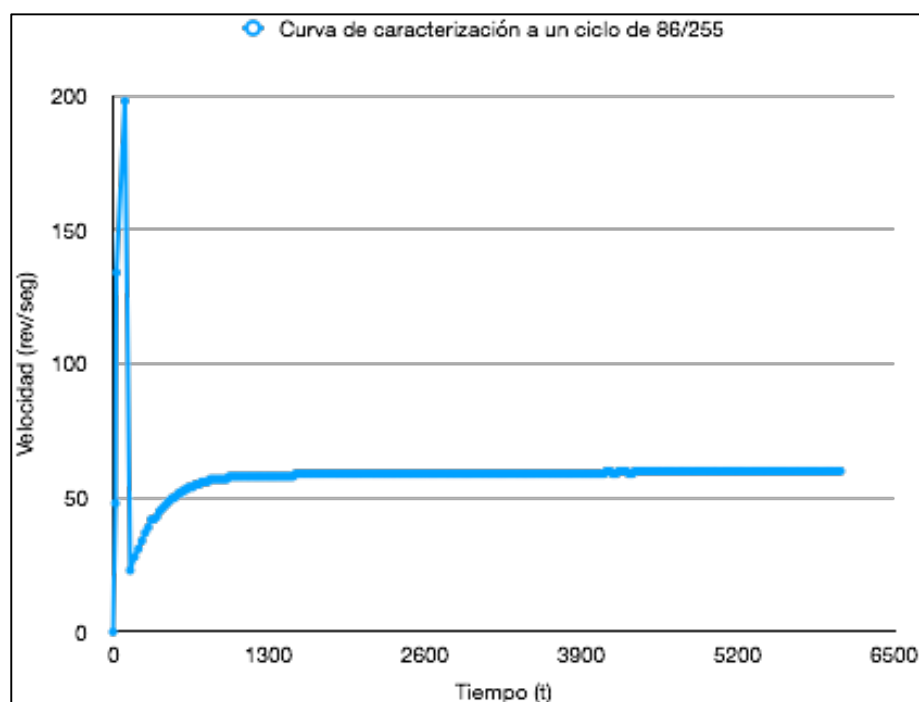


Figura 12: Curva de caracterización del motor a un ciclo de 85/255

1.2. Control PID de un motor de CC.

Se repiten los pasos 1 – 6 de la caracterización del motor de CC:

1. “Abrir System Workbench for STM32” e ir al fichero main.c
2. Compilar el programa.
3. Corregir los errores expuestos.
4. Abrir el software PuTTY.
5. Configurar la comunicación entre la tarjeta STM32F411E Discovery y la PC.
6. Clic en el botón azul de la tarjeta STM32F411E Discovery.
7. Realizar el control de velocidad mediante un controlador PID (menú 2).

```
Por favor indique el tipo de actividad:
1. Prueba motor
2. Control PID
3. Control basico
4. Caracterizar motor y guardar Info dinamica
5. Cambiar parametros PID
Modo control
Valor P= 1280
Valor I= 10
Valor D= 0
Indique la velocidad deseada y luego presione la tecla E (enviar). La tecla A (a
bortar) permite salir del proceso de control
60Estableciendo velocidad 60
Vel.Des:60 Vel.Med.:60 Vel.Prom:0 Acc.Cont:81 Error:0 Ciclo:96 eInt: 2092
Vel.Des:60 Vel.Med.:60 Vel.Prom:0 Acc.Cont:79 Error:0 Ciclo:94 eInt: 2041
Vel.Des:60 Vel.Med.:60 Vel.Prom:0 Acc.Cont:77 Error:0 Ciclo:92 eInt: 1992
Vel.Des:60 Vel.Med.:60 Vel.Prom:0 Acc.Cont:76 Error:0 Ciclo:91 eInt: 1958
Vel.Des:60 Vel.Med.:60 Vel.Prom:0 Acc.Cont:75 Error:0 Ciclo:90 eInt: 1941
Vel.Des:60 Vel.Med.:60 Vel.Prom:0 Acc.Cont:75 Error:0 Ciclo:90 eInt: 1924
Vel.Des:60 Vel.Med.:60 Vel.Prom:0 Acc.Cont:74 Error:0 Ciclo:89 eInt: 1908
Vel.Des:60 Vel.Med.:60 Vel.Prom:0 Acc.Cont:73 Error:0 Ciclo:88 eInt: 1891
```

Figura 13: Menú control PID

Esta opción permite realizar el control PID del motor, siendo las constantes:

- Proporcional = 1280.
- Integral = 10.
- Derivativa = 0, los valores iniciales.

Al teclear una velocidad (en este caso 60 rev/seg), el microcontrolador realiza una serie de operaciones y rutinas según la velocidad medida para tratar de obtener el valor deseado.

Cuando se ha establecido una velocidad, las interrupciones nuevamente han sido activadas y el programa intenta controlar el motor para que gire a la velocidad deseada, mediante la medición de parámetros obtenidos por el optoacoplador. Dichos valores son enviados a los controladores PID y asignará el ciclo correcto correspondiente de PWM.

Analizando la figura 13:

- Se presenta, el valor de la velocidad deseada antes tecleada.
- Valor de la velocidad medida por el programa.
- Velocidad promedio durante el último segundo.
- Acc.Cont es el valor del cálculo del PID.
- Error es la diferencia actual entre la velocidad deseada y la velocidad medida.
- Ciclo es el valor del tiempo activo del PWM (número entre 0 y 255).
- eInt es el valor de la variable acumuladora del error que usa el control integral.

Al pulsar la tecla A se mostrará las velocidades medidas durante los tres primeros segundos de iniciar el control PID y se regresará al menú principal. Estos datos pueden ser exportados para luego introducirlos en Microsoft Excel. Al final se puede genera una curva de velocidad en donde se puede apreciar la respuesta al escalón del control PID.

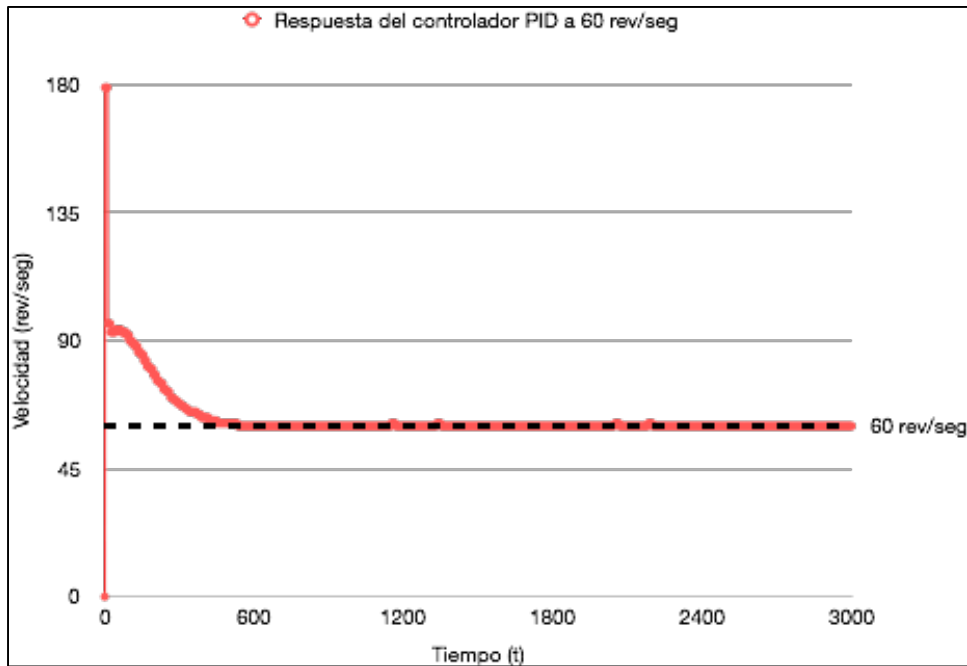


Figura 14: Respuesta del motor a una velocidad de 60 rev/seg

8. Cambiar las constantes del PID es el último paso que ejecuta el estudiante (menú 5).

```

Por favor indique el tipo de actividad:
1. Prueba motor
2. Control PID
3. Control basico
4. Caracterizar motor y guardar Info dinamica
5. Cambiar parametros PID
Modo cambiar valores PID (SE DIVIDEN POR 2^8). Se cambian uno a uno cuando se pu
lsa E se pasa al siguiente
Valor P=
1536
Valor I=
12
Valor D=
0
Parametro P cambio de 1280/256 a valor 1536/256
Parametro I cambio de 10/256 a valor 12/256
Parametro D cambio de 0/256 a valor 0/256
  
```

Figura 15: Menú cambiar parámetros PID

Se había comentado que los valores iniciales de las constantes son: $P=1280$, $I=10$, $D=0$. Se debe teclear el valor de la constante proporcional, integral y derivativa (un valor multiplicado por 256) (Profesores del Dpto. de Ingeniería Electrónica de la ETSI Telecomunicación de la UPM, 2017).

Una vez que se han cambiado de los valores PID, estos nuevos parámetros se mostraran en pantalla, se regresa al menú principal y el controlador quedará modificado.

2. Filtrado de datos adquiridos

En cualquiera de las prácticas 1.2 y 1.3 se han adquirido una serie de datos tal y como muestra la siguiente figura:

```
*****
Tiempos y Muestras almacenadas
INICIO
0,0
    20,48
221,29
282,33
338,37
390,39
439,41
509,44
553,45
```

Figura 16: Muestras almacenadas

Se ha implementado un código en el lenguaje de programación PERL. Dicho programa filtra la información mostrada en la pantalla de PuTTY.

- Para guardar la sesión, se da clic derecho en la barra superior cerca de los botones “- [] X” de PuTTY y se selecciona la opción “copy All to Clipboard”.

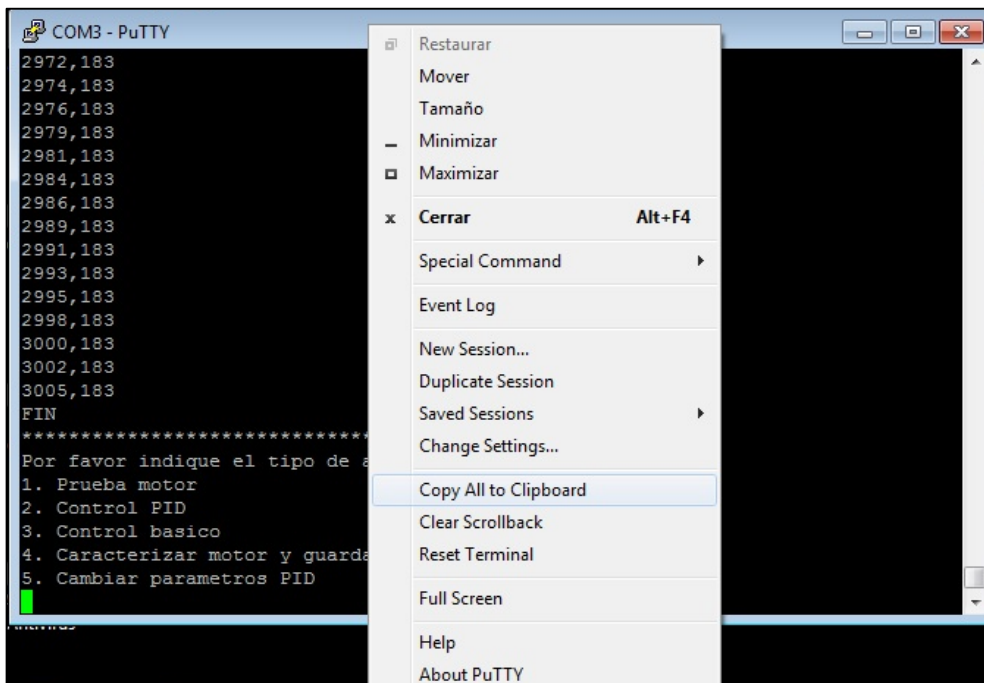


Figura 17: Guardar sesión PuTTY

1. Realizado esto, el usuario tiene que ubicarse en la siguiente dirección: TFM/Pruebas_MOTOR, abrir el editor de texto, opción “pegar” y guardar con el nombre “fichero.txt”.

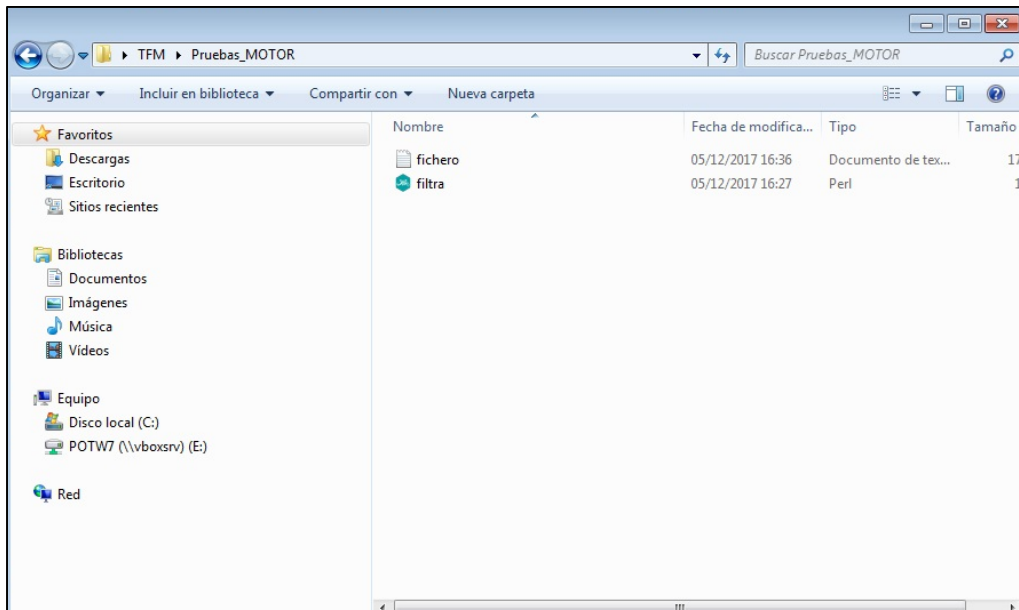


Figura 18: Carpeta Pruebas_MOTOR

2. Luego, se abre el símbolo de sistema (Menú de Windows/ejecutar/cmd).
3. Ingresar el siguiente código:
 - Cd Desktop/TFM/Pruebas_MOTOR
4. Por último, se ejecuta la línea:
 - perl filtra.pl < fichero.txt

Si todo sale correctamente, el programa “filtra” crea los ficheros en base a las pruebas realizadas:

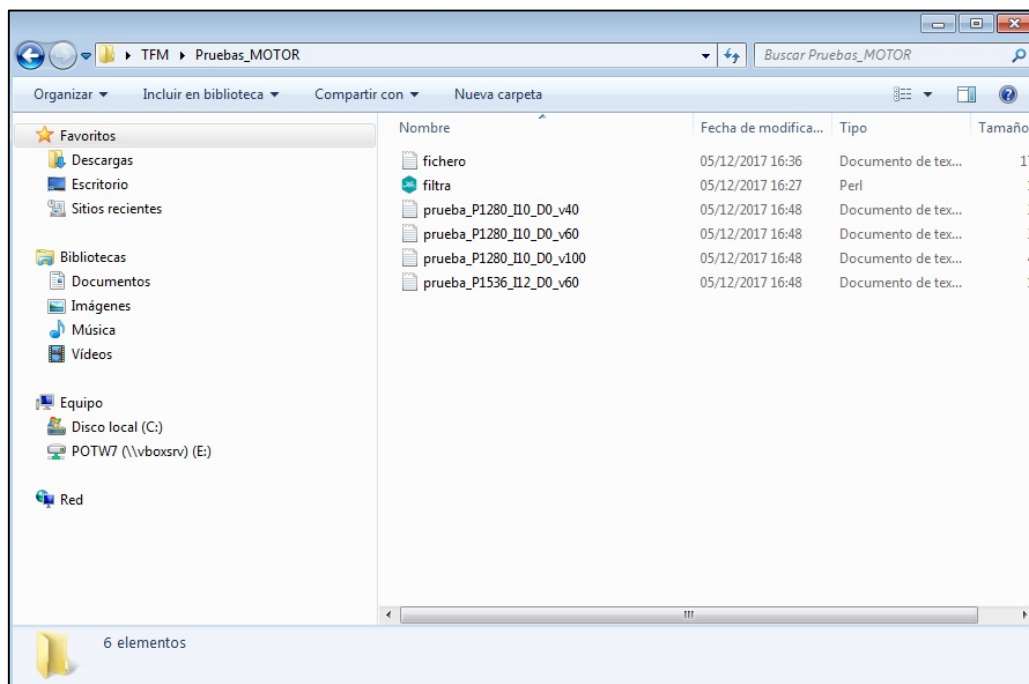


Figura 19: Carpeta Pruebas_MOTOR con ficheros de pruebas

Los ficheros creados tendrán el formato:

1. Si se ha realizado pruebas en el menú 2: Control PID:
 - prueba_PIXX_IXX_DXX_vXX, donde las “XX” serán remplazadas con el valor de las constantes PID y la velocidad deseada.
2. Si se ha realizado pruebas en el menú 4: Caracterización:
 -

EL PROGRAMA EN PERL FILTRA LAS PRUEBAS REALIZADAS EN EL MENU 2, PERO NO EN EL MENU 4.

Conclusiones

- Una vez que se ha cumplido con la ejecución correcta de las prácticas 1.2 y 1.3, el alumno es capaz de caracterizar un motor de CC y controlar su velocidad, utilizando un control PID, mediante la tarjeta STM32F411E Discovery.
- Se ha citado que el algoritmo implementado permite controlar el sentido de giro del motor. Esto no se ha ejecutado del todo; ya que, por el momento al inicializar los periféricos, se está eligiendo un solo sentido. En futuras líneas se ha propuesto que el estudiante pueda elegir mediante el teclado el sentido de giro del motor.
- Para realizar una prueba de sentido de giro, el usuario debe ir al fichero turn.c; comentar la línea 22 y quitar comentarios en la línea 21. ¿Qué se ha hecho?, la GPIO PC0 es la encargada del sentido de giro; en la función __init() se ha llamado a la función MX_GPIO_Init() que configura dicho puerto y coloca un “0” lógico por una sola vez. Si el valor se ha cambiado por “GPIO_PIN_SET”, el motor gira en el otro sentido.
- El menú 3 (control básico) no es una opción recomendable para realizar el control de velocidad del motor, debido a que es un controlador simple y muy lento. ¿Qué hace?, se ingresa la velocidad a controlar (por ejemplo 70 rev/seg), inicializa el PWM y compara la velocidad medida con la deseada y cada 1 seg aumenta (o disminuye) un ciclo hasta que el error sea igual a 0.
- La tarjeta STM32F411E Discovery es un sistema muy completo; además de tener los periféricos que se utilizaron en el desarrollo de estas prácticas, posee herramientas adicionales como: acelerómetro, giroscopio, acceso, lectura y escritura de memoria, entrada y salida de audio y demás componentes que pueden ser utilizados para futuras prácticas de control de procesos.

Recomendaciones

1. Debido a que las prácticas, ejecución del IDE y comunicación serial se realiza en la máquina virtual, se puede presentar error en la conexión de la tarjeta STM32F411E Discovery. Se sugiere que los casilleros de “STMicroelectronics STM32 XXX” estén señalados; de manera que estén “conectados” al Windows 7 virtual, tal y como se muestra en la siguiente figura:

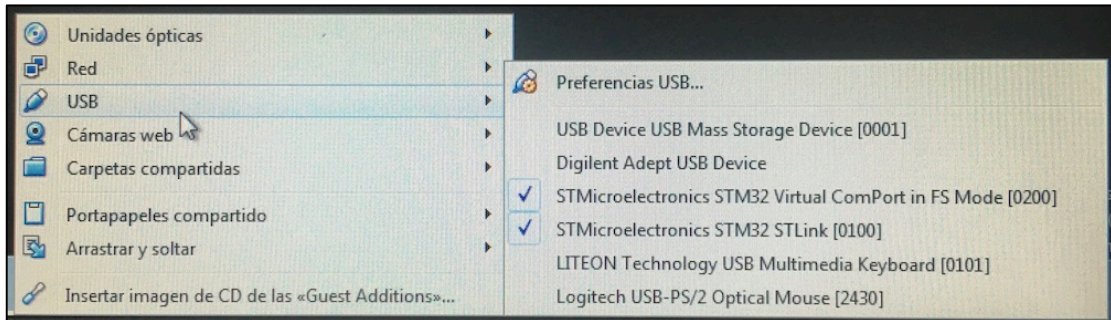


Figura 20: Conexión correcta del periférico USB a máquina virtual

2. Se ha añadido un filtro en la conexión USB de la máquina virtual, de manera que, cada vez que se conecte la tarjeta STM32F411E Discovery al puerto USB, se “conecte” automáticamente al Windows 7 virtual; sin embargo, se puede presentar el siguiente error a la hora de cargar el programa en el sistema empujado:

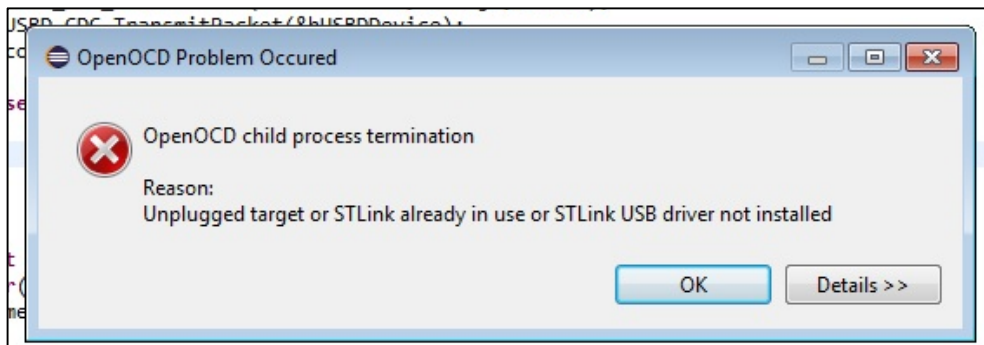


Figura 21: Error de conexión con la tarjeta STM32F411E Discovery

¿Qué sucedió?, al momento de cargar un programa, la tarjeta STM32F411E Discovery se reinicia automáticamente (por ese motivo, se desconecta de la PC por un instante) y se pierde la conectividad. Una solución es desconectar la tarjeta de la PC y volver a conectar.

3. La comunicación USB es fundamental para estas prácticas y la virtualización del puerto serial varía entre una PC y otra. Para saber que puerto COM es el correcto a la hora de configurar PuTTY, se tiene:

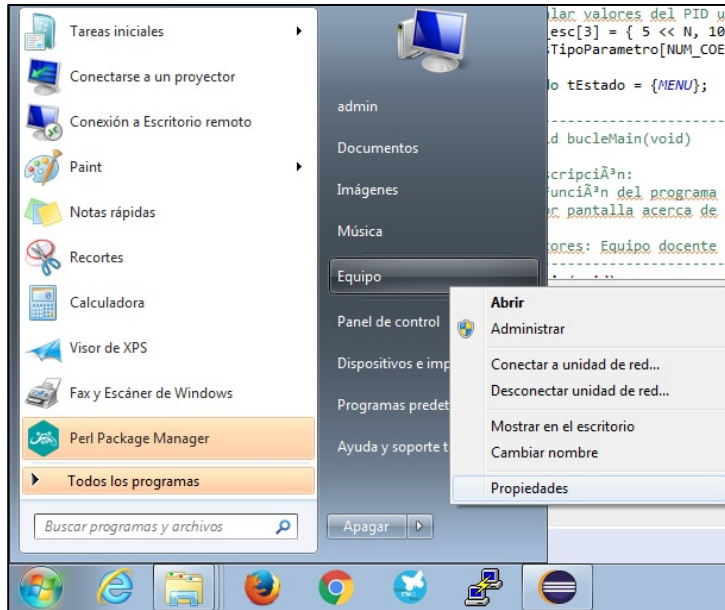


Figura 22: Propiedades del sistema de Windows 7

4. Dentro de esta opción se busca “Administrador de dispositivos” y se despliega la opción “Puertos (COM y LPT)”:

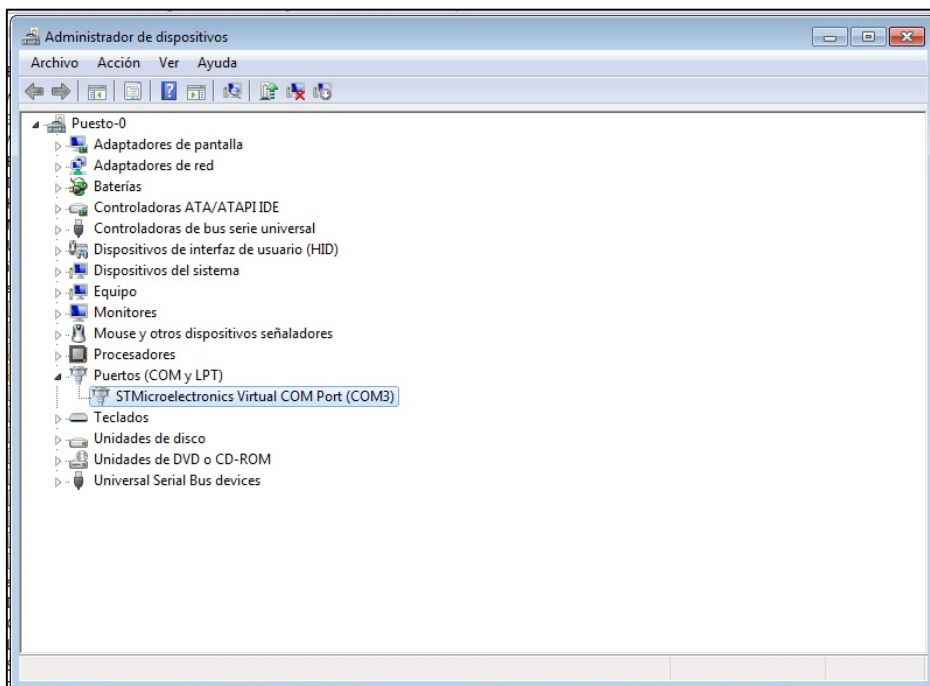


Figura 23: Administrador de dispositivos de Windows 7

5. Si la opción de la figura 23 no está disponible, intentamos solucionar con la recomendación 1.

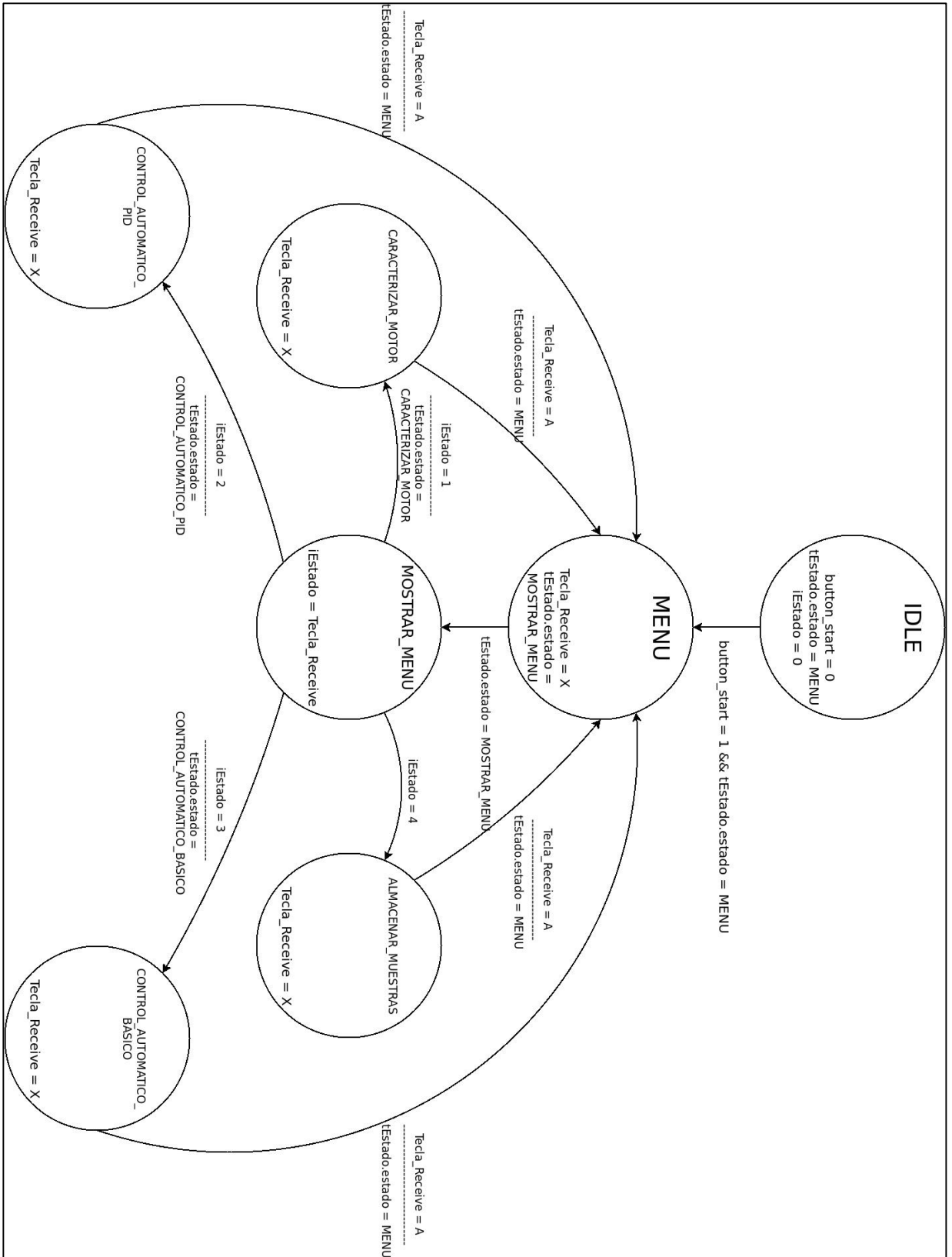
Una herramienta muy eficiente es la documentación disponible en la página web de STMicroelectronics, ya que se puede encontrar el datasheet de la tarjeta, manual de usuario, manual de referencia y manual de la librería HAL; así como demás periféricos disponibles.

Bibliografía

- Buestán, P. A. (1 de 2018). Desarrollo de prácticas de control utilizando la tarjeta Discovery STM32 Discovery. Madrid, Madrid, Spain.
- Malagón Marzo, P., & Fernández, F. (1 de 1 de 2017). Laboratorio de Sistemas Electrónicos - MUISE (2016-2017). Madrid, Spain.
- NXP. (1 de 12 de 1990). *74HC/HCD244*. Recuperado el 22 de 11 de 2017, de ALLDATASHEET.ES: <http://pdf1.alldatasheet.es/datasheet-pdf/view/15571/PHILIPS/74HC244.html>
- NXP. (1 de 09 de 1993). *74HC/HCT14*. Recuperado el 22 de 11 de 2017, de ALLDATASHEET.ES: <http://pdf1.alldatasheet.es/datasheet-pdf/view/15537/PHILIPS/74HC14.html>
- NXP. (1 de 1 de 2007). *74HC240; 72HCT240*. Recuperado el 22 de 11 de 2017, de ALLDATASHEET.ES: <http://www.alldatasheet.es/datasheet-pdf/pdf/211060/NXP/74HC240.html>
- Profesores del Dpto. de Ingeniería Electrónica de la ETSI Telecomunicación de la UPM. (1 de 05 de 2017). Práctica de control con el MCF5272: caracterización de un motor de corriente continua. Madrid, Spain.

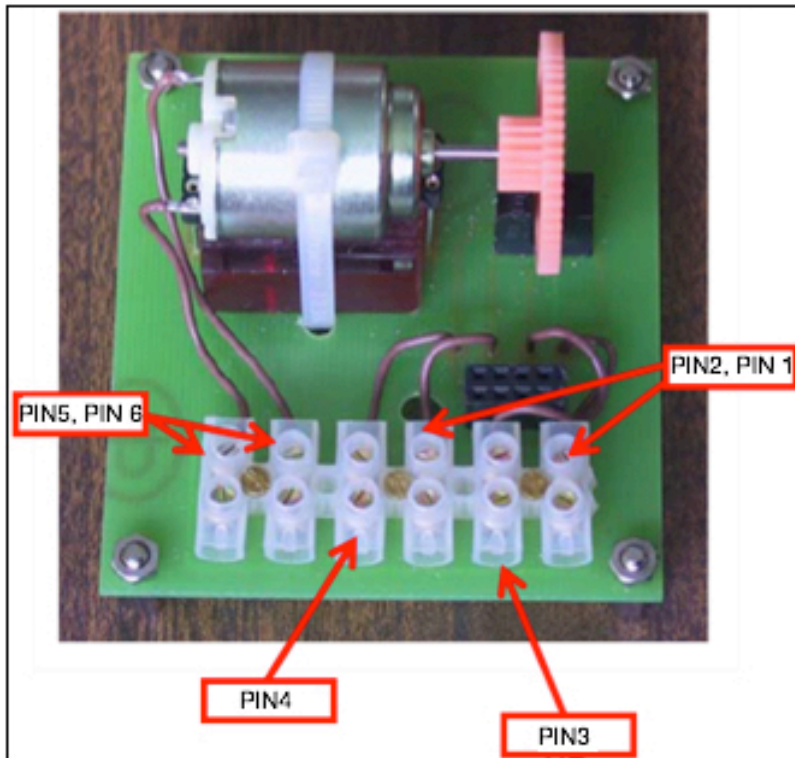
Anexos

- Máquina de estados (menú).

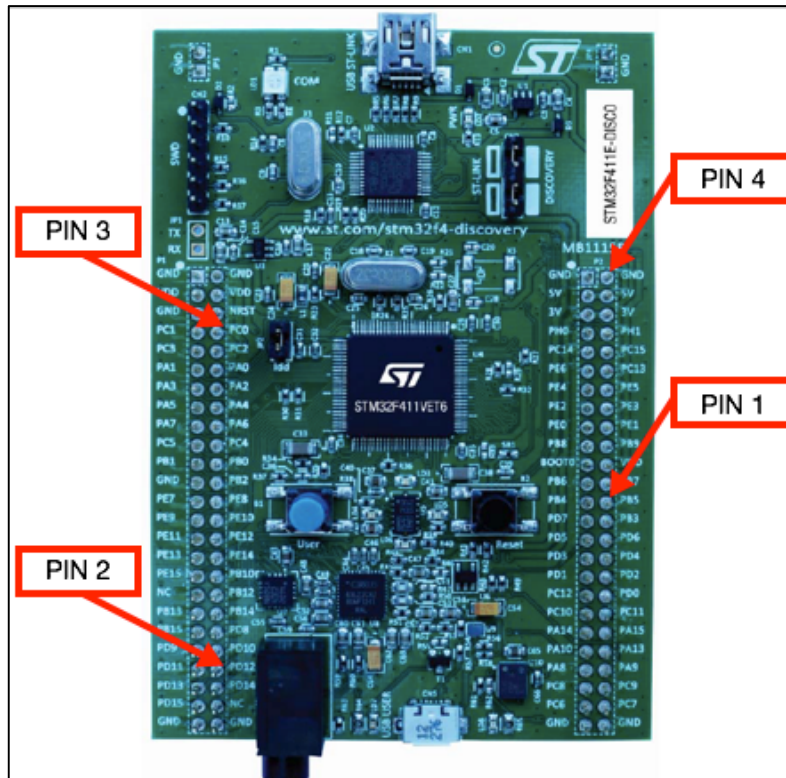


- Conexión de la tarjeta STM32F411E Discovery y el motor de CC con las ranuras de entrada/salida.

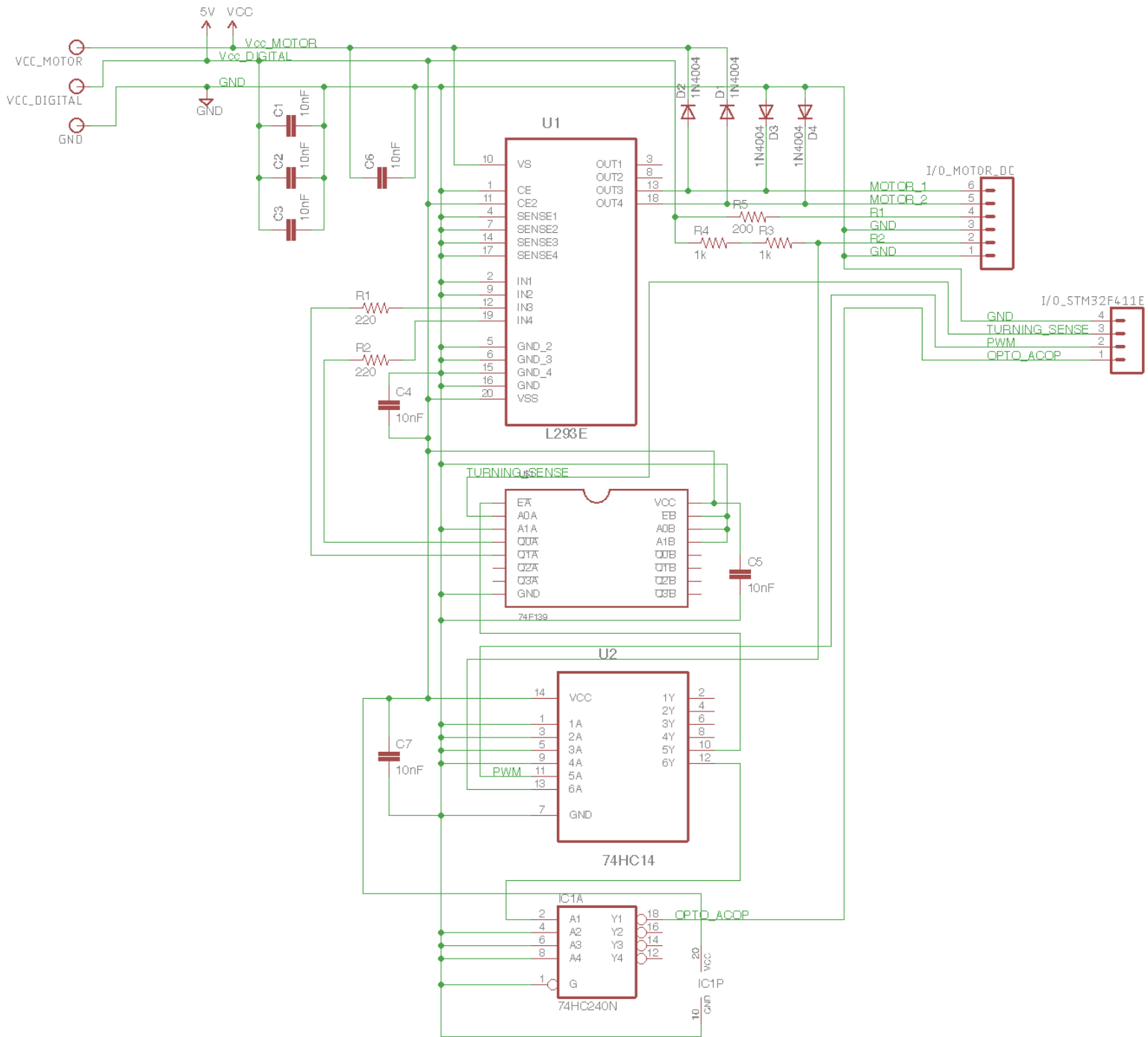
I/O MOTOR_DC



I/O_STM32



- Circuito esquemático completo (hardware).



- Circuito layout completo (hardware).

