

SISTEMAS ELECTRÓNICOS DIGITALES

TRABAJO MICROS

CONTROL DE PERSIANAS



POLITÉCNICA

ALUMNOS:

Sara Rodríguez Fernández 54832

Víctor Rodríguez Sánchez 54838

Oscar Rudek 54844

FECHA: 19 de enero de 2022

PROFESOR: Luis Castedo

INDICE

1. INTRODUCCIÓN	3
2. DESCRIPCION Y DESARROLLO DEL TRABAJO	3
2.1 Componentes	3
2.2 Descripción de la estrategia y proyecto	3
2.3 Algoritmos desarrollados	3
3. DIAGRAMAS	4
3.1 Bloque de estados	4
3.2 Esquema global del proyecto	5
3.3 Esquemas componentes según funcionabilidad y conexiones	5
4. EXPLICACIÓN DE LAS FUNCIONES Y DE SU INTERFAZ	7
Función: INICIALIZACIÓN DE VARIABLES Y PARÁMETROS	7
Función: While: ESTADO MANUAL y ESTADO AUTOMÁTICO	7
Función: GESTIÓN DE LAS INTERRUPCIONES	8
Función: CONTROL DEL SERVOMOTOR	8
Función: GESTIÓN MOVIENTO DE LA PERSIANA	9
Función: GESTIÓN DE LOS MENSAJES ENVIADOS EN MODO AUTO.....	10
Función: Control pantalla LCD1602	10

1. INTRODUCCIÓN

Nuestro trabajo se basa en el control de una persiana mediante los modos manual y automático. El modo manual depende de pulsadores y el modo automático depende de la luz que recibe la fotoresistencia del exterior. De esta forma, ambos modos accionan el servo de formas diferentes.

2. DESCRIPCION Y DESARROLLO DEL TRABAJO

2.1 Componentes

- STM32F407G – DISC1
- Protoboard
- Servomotor SG90
- LCD sin comunicación I2C 16 x 2
- Potenciómetro
- Resistencias 1 k Ω
- LDR (fotoreistencia)
- Botones
- Cables

2.2 Descripción de la estrategia y proyecto

La estrategia a seguir es la creación de una máquina de dos estados, el estado de manual y el estado de automático. Ambos estados se regulan mediante las entradas que son gestionadas mediante interrupciones accionadas por los botones del montaje y de la placa.

El modo manual permite subir y bajar la “persiana” con la ayuda de dos botones que accionan un servomotor, uno para bajarla y otro para subirla. En el modo automático se encarga la fotoresistencia del movimiento de la “persiana”, dependiendo de la cantidad de luz recibida y del umbral establecido la “persiana” subirá o bajará.

Además, se cuenta con la integración de una pantalla LCD1602 para poder monitorizar cada estado, cada movimiento y cada interfaz.

2.3 Algoritmos desarrollados

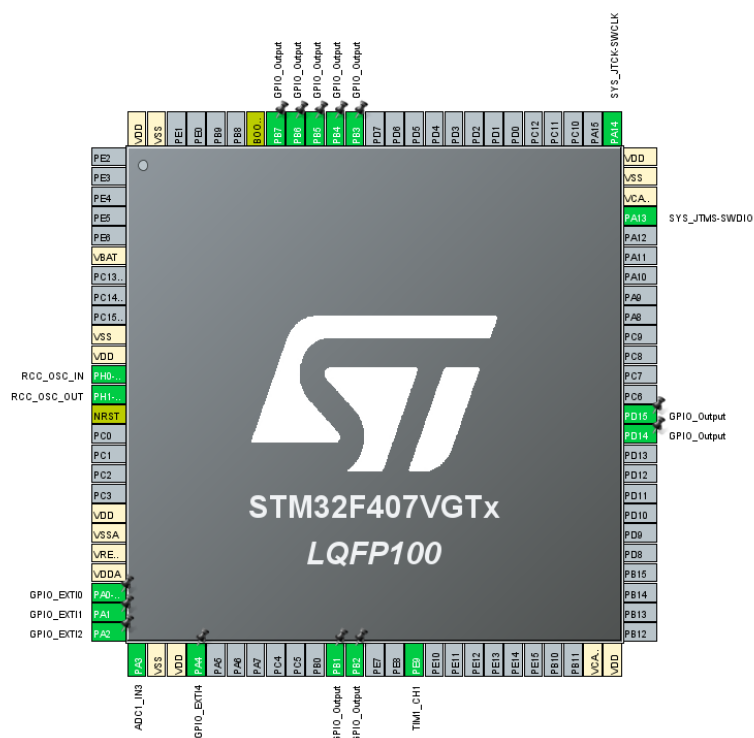
Utilización de entradas con interrupciones para los botones exteriores y el propio botón de la placa que ayudan a la rápida activación del siguiente estado o etapa. Además, usamos los leds de la placa para mostrar el final de una acción como salidas del sistema.

Hemos usado dos temporizadores de la placa. El TIM1 y el TIM4. El TIM1 se encarga del control del servo generando una señal de 50 Hz para su control y el TIM4 se encarga del control de la pantalla LCD1602 con una señal aproximada de 15 Hz. La LCD1602 establece una comunicación paralela con la placa que nos permite mandarle mensajes para que los muestre por pantalla.

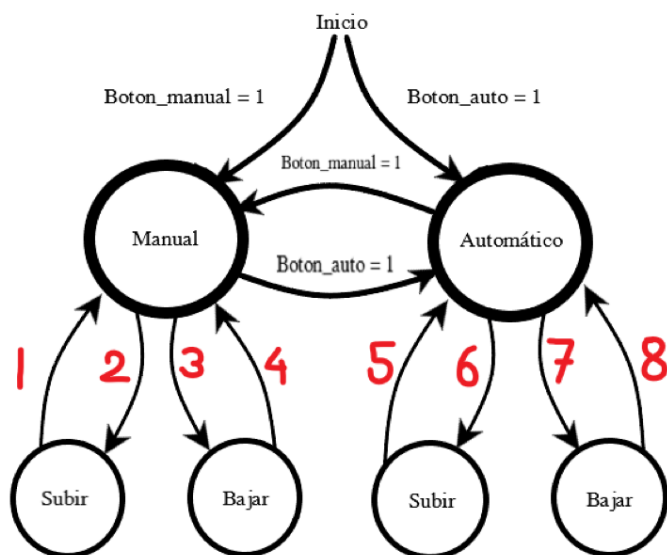
También hemos incorporado un puerto como convertidor analógico/digital para la lectura de la LDR y poder trabajar con los datos leídos.

3. DIAGRAMAS

A continuación se muestra el diagrama global de la placa configurada en el archivo .ioc del proyecto:

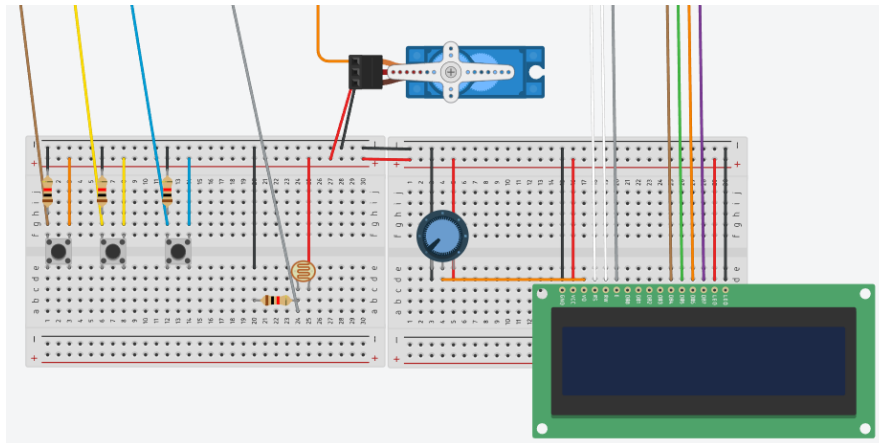


3.1 Bloque de estados



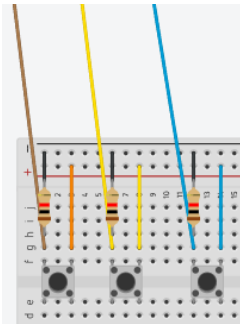
- 1) !subir_ini && subir_fin && !bajar_fin
- 2) subir_ini && !subir_fin
- 3) bajar_ini && !bajar_fin
- 4) !bajar_ini && bajar_fin && !subir_fin
- 5) subir_fin && !bajar_fin
- 6) (valor_luminosidad > UMBRAL_LUMINOSIDAD_ALTA) && !subir_fin
- 7) (valor_luminosidad < UMBRAL_LUMINOSIDAD_BAJA) && !bajar_fin
- 8) bajar_fin && !subir_fin

3.2 Esquema global del proyecto



3.3 Esquemas componentes según funcionalidad y conexiones

Botones con interrupciones:

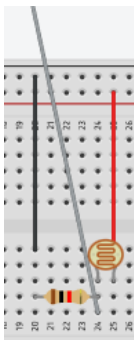


El cable azul está conectado al puerto PA1 y está asignado la función de activar el modo manual.

El cable amarillo está conectado al puerto PA2 y está asignado la función de bajar la persiana en modo manual.

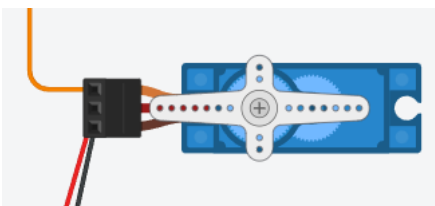
El cable marrón está conectado al puerto PA4 y está asignado la función de subir la persiana en modo manual.

LDR Convertidor A/D



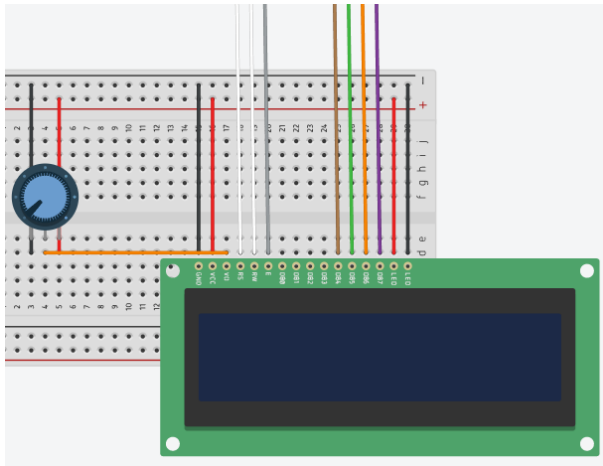
El cable gris está conectado al puerto PA3 que es el encargado de transformar la luz recibida por la LDR.

Servomotor controlado por el temporizador



El cable naranja está conectado al puerto PE9 que es el canal 1 del temporizador 1.

Pantalla LCD1602



Realiza una comunicación en paralelo con la placa. Está conectada a un potenciómetro para regular la intensidad de los leds.

Las conexiones se realizan el puerto B de PB1 hasta PB7

4. EXPLICACIÓN DE LAS FUNCIONES Y DE SU INTERFAZ

Función: INICIALIZACIÓN DE VARIABLES Y PARÁMETROS

```

240  /* USER CODE BEGIN 2 */
241
242  /*-----INICIALIZACION DE VARIABLES Y PARÁMETROS-----*/
243  HAL_TIM_Base_Start(&htim4);
244  lcd_init();
245  lcd_clear();
246  HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_1);
247  htim1.Instance->CCR1 = 25;
248  bajar_fin = 1;
249  HAL_Delay(100);
250  HAL_TIM_PWM_Stop(&htim1, TIM_CHANNEL_1);
251  HAL_Delay(100);
252
253  /* USER CODE END 2 */

```

Lo primero es establecer las condiciones de inicio del proyecto. Para ello, iniciamos la pantalla LCD1602 y borramos su contenido por si se ha quedado algo guardado en ella, después iniciamos el temporizador 1 para establecer el servo en la posición de bajada, activamos la flag de que se encuentra bajada la persiana y paramos el temporizador.

Función: While: ESTADO MANUAL y ESTADO AUTOMÁTICO

```

257  while (1)
258  {
259      /* USER CODE END WHILE */
260
261      /* USER CODE BEGIN 3 */
262      /*-----ESTADO MANUAL-----*/
263      if (boton_manual){
264          lcd_put_cur(0, 0);
265          lcd_send_string("                ");
266          lcd_put_cur(1, 0);
267          lcd_send_string("MODO: MANUAL");
268
269          if (subir_ini && !subir_fin){
270              lcd_clear();
271              subirpersiana();
272          }
273
274          if (bajar_ini && !bajar_fin){
275              lcd_clear();
276              bajarpersiana();
277          }
278      }
279
280      /*-----ESTADO AUTOMÁTICO-----*/
281      if (boton_auto){
282          lcd_put_cur(0, 0);
283          lcd_send_string("MODO: AUTOMATICO");
284          lcd_put_cur(1, 0);
285          lcd_send_string("                ");
286
287          HAL_ADC_Start(&hadc1);
288
289          if (HAL_ADC_PollForConversion(&hadc1, HAL_MAX_DELAY) == HAL_OK){
290              adc_val_luminosidad = HAL_ADC_GetValue(&hadc1);
291
292              if ((adc_val_luminosidad > UMBRAL_LUMINOSIDAD_ALTA) && !subir_fin){
293                  lcd_clear();
294                  subirpersiana();
295                  mensaje_buenos_dias();
296              }
297
298              if ((adc_val_luminosidad < UMBRAL_LUMINOSIDAD_BAJA) && !bajar_fin) {
299                  lcd_clear();
300                  bajarpersiana();
301                  mensaje_buenos_noches();
302              }
303          }
304
305          HAL_ADC_Stop(&hadc1);
306      }
307  }
308  /* USER CODE END 3 */
309

```

Este bucle while se ejecuta indefinidamente, dependiendo de si se ha activado el modo manual o el automático entrará en una función u otra.

En modo manual depende del botón que haya pulsado para subir o bajar la persiana se ejecutará la función de subirpersiana o de bajarpersiana. Además, hay que tener en cuenta de los avisos de subir_fin y bajar_fin para saber la posición de la persiana y bloquear por ejemplo que si la persiana está bajada se active el servo al pulsar bajar.

En modo automático se inicia el convertido A/D para que empiece a tomar valores, dependiendo del valor tomado entrará en un umbral u otro, o simplemente se queda tomando valores hasta llegar a los de los umbrales. Dependiendo del umbral que pase se llama a una función u otra.

Cabe destacar que la pantalla LDC1602 muestra todo el rato el modo actual y las interacciones que se van realizando.

Función: GESTIÓN DE LAS INTERRUPCIONES

```

86  /*-----GESTIÓN DE LAS INTERRUPCIONES PRODUCIDAS POR LOS BOTONES-----*/
87  void HAL_GPIO_EXTI_Callback(uint16_t GPIO_Pin){
88
89      if (GPIO_Pin==GPIO_PIN_0){          /*COMPROBACIÓN DE LA ACTIVACIÓN DEL MODO AUTOMÁTICO*/
90          boton_auto = 1;
91          boton_manual = 0;
92          subir_ini = 0;
93          bajar_ini = 0;
94      }
95
96      else if(GPIO_Pin==GPIO_PIN_1){      /*COMPROBACIÓN DE LA ACTIVACIÓN DEL MODO MANUAL*/
97          boton_auto = 0;
98          boton_manual = 1;
99          subir_ini = 0;
100         bajar_ini = 0;
101     }
102
103     else if(GPIO_Pin==GPIO_PIN_2){      /*GESTIÓN DEL BOTON EXTERNO PARA BAJAR LA PERSIANA*/
104         if (boton_manual){
105             boton_auto = 0;
106             boton_manual = 1;
107             subir_ini = 0;
108             bajar_ini = 1;
109         }
110     }
111
112     else if(GPIO_Pin==GPIO_PIN_4){      /*GESTIÓN DEL BOTON EXTERNO PARA SUBIR LA PERSIANA*/
113         if (boton_manual){
114             boton_auto = 0;
115             boton_manual = 1;
116             subir_ini = 1;
117             bajar_ini = 0;
118         }
119     }
120 }
121
122
123

```

Esta función es la encargada del control de las variables “banderas” que se activan con las interrupciones externas de los botones.

Función: CONTROL DEL SERVOMOTOR

```

165  /*-----GESTIÓN DE CONTROL UN SERVO PARA LA SUBIDA-----*/
166  int servo_girar_up(int ini, int fin){
167
168      while(ini<=fin){
169          htim1.Instance->CCR1 = ini;
170          ini++;
171          HAL_Delay(25);
172      }
173      return 1;
174  }
175
176  /*-----GESTIÓN DE CONTROL UN SERVO PARA LA BAJADA-----*/
177  int servo_girar_down(int ini, int fin){
178
179      while(ini>=fin){
180          htim1.Instance->CCR1 = ini;
181          ini--;
182          HAL_Delay(25);
183      }
184      return 1;
185  }

```


Para trabajar con el servo es necesario generar ondas de 50Hz (20ms), pero para poder controlar el servo es necesario asignarle una señal de ancho de pulso entre 0.5 ms y 2.5 ms. Con la configuración del temporizador 1 el periodo es 1000 por lo que para asignarle 0.5ms de PWM hay que asignarle en su comparador el valor 25 y para una onda de 2.5 ms asignarle un valor de 125 en el comparador. Por lo que asignando 25 a CCR1 estaremos en 0° y asignando 125 en 180°.

En estas funciones lo que se hace es asignar el valor inicial de 25 para subir la persiana o de 125 para bajar la persiana y dependiendo de la función aumentar de uno en uno hasta 125 en caso de subirla o disminuir de uno en uno el valor hasta 25 para bajarla.

Función: GESTIÓN MOVIMIENTO DE LA PERSIANA

```

125 /*-----GESTIÓN DEL MOVIMIENTO DE LA "PERSIANA" DE SUBIR MEDIANTE UN SERVO-----*/
126 void subirpersiana(){
127     lcd_put_cur(0, 4);
128     lcd_send_string("SUBIENDO...");
129     HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_1);
130     servo_fin_up = servo_girar_up(25,125);
131     HAL_Delay(500);
132     lcd_clear();
133     if (servo_fin_up){
134         HAL_TIM_PWM_Stop(&htim1, TIM_CHANNEL_1);
135         HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_SET); /*ACTIVACIÓN PARA SEÑALIZAR QUE LA PERSIANA HA TERMINADO DE BAJAR*/
136         HAL_Delay(2000);
137         HAL_GPIO_WritePin(GPIOD, GPIO_PIN_14, GPIO_PIN_RESET);
138         subir_ini = 0;
139         subir_fin = 1;
140         bajar_fin = 0;
141         servo_fin_up = 0;
142     }
143 }
144
145 /*-----GESTIÓN DEL MOVIMIENTO DE LA "PERSIANA" DE BAJAR MEDIANTE UN SERVO-----*/
146 void bajarpersiana(){
147     lcd_put_cur(0, 4);
148     lcd_send_string("BAJANDO...");
149     HAL_TIM_PWM_Start(&htim1, TIM_CHANNEL_1);
150     servo_fin_down = servo_girar_down(125,25);
151     HAL_Delay(500);
152     lcd_clear();
153     if (servo_fin_down){
154         HAL_TIM_PWM_Stop(&htim1, TIM_CHANNEL_1);
155         HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_SET); /*ACTIVACIÓN PARA SEÑALIZAR QUE LA PERSIANA HA TERMINADO DE SUBIR*/
156         HAL_Delay(2000);
157         HAL_GPIO_WritePin(GPIOD, GPIO_PIN_15, GPIO_PIN_RESET);
158         bajar_ini = 0;
159         subir_fin = 0;
160         bajar_fin = 1;
161         servo_fin_down = 0;
162     }
163 }

```

Empezando por la función subirpersiana, se muestra por la pantalla la acción de subiendo... Se inicia el temporizador que controla el servo y se llama a la función que incrementa el valor del comparador del temporizador en modo PWM pasando el valor inicial de 25 y final de 125, cuando el comparador ha alcanzado su valor máximo devuelve un uno que se le asigna a la variable de control servo_fin_up. Cuando se activa esta variable se para el temporizador y se enciende uno de los leds de la placa para mostrar el fin de la acción de subir y se apaga después de dos segundos. Como se observa se asignan los valores correspondientes a las variables de control y se sale de la función.

Para la función bajarpersiana es al contrario, la pantalla LCD1602 muestra la acción de bajando, se inicia el temporizador que controla el servo y se llama a la función que disminuye el valor del comparador del temporizador en modo PWM pasando valor inicial 125 y final 25. Cuando se llega al valor mínimo se devuelve un uno a la variable de control servo_fin_down y se realiza un ciclo igual ala anterior función. Se apaga el temporizador y se enciende un led de la placa para simbolizar el final de la acción. Finalmente se asigna su valor correspondiente a las variables de control.

Función: GESTIÓN DE LOS MENSAJES ENVIADOS EN MODO AUTO

```
187  /*-----GESTIÓN DE MENSAJE 1-----*/
188 void mensaje_buenos_dias(void){
189     lcd_put_cur(0, 2);
190     lcd_send_string("BUENOS DIAS");
191     lcd_put_cur(1, 5);
192     lcd_send_string("ARRIBA");
193     HAL_Delay(4000);
194     lcd_clear();
195 }
196
197  /*-----GESTIÓN DE MENSAJE 2-----*/
198 void mensaje_buenos_noches(void){
199     lcd_put_cur(0, 1);
200     lcd_send_string("BUENAS NOCHES");
201     lcd_put_cur(1, 4);
202     lcd_send_string("A DORMIR");
203     HAL_Delay(4000);
204     lcd_clear();
205 }
```

Estas funciones son llamadas en el modo automático para mandar un mensaje diferente al bajar o subir la persiana.

Función: Control pantalla LCD1602

Para el manejo de la pantalla LCD1602 usamos las funciones:

- lcd_init() → para la inicialización de la misma
- lcd_clear() → para limpiar su contenido
- lcd_put_cur(int, int) → para establecer el cursor en la posición deseada
- lcd_send_string("") → para mandar el mensaje que se desea mostrar

Todas las funciones para el control de la pantalla LCD1602 se encuentran en el archivo LCD1602.c