Universidad de Buenos Aires
Facultad de Ingeniería

86.07 Laboratorio de Microprocesadores

Trabajo Práctico N°1: Manejo de Puertos

Díaz Falvo, Malena - 102374

Curso: Martes - Segundo Cuatrimestre de 2020

Fecha de entrega: 27/10/20

Docentes:
Gerardo Stola
Fernando Cofman (corrector)
Guido Salaya

Índice

1. Objetivos

El principal propósito de este trabajo práctico es familiarizarse con el manejo de registros de puertos en un microcontrolador ATMega328P. Además, se busca observar la utilidad de la resistencia de *pull-up* interna que poseen dichos dispositivos.

2. Descripción

El proyecto está compuesto por tres prácticas a realizar. La primera consistió en hacer parpadear un diodo LED conectado al PIN 2 del microcontrolador, utilizando como base una rutina de retardo provista por la cátedra.

Según la segunda consigna, se modificó el programa implementado para que el LED parpadeara cuando se presione el pulsador 1 y se apague al apretar el pulsador 2. El circuito implementado se tomó según el esquemático brindado por la cátedra expuesto en la figura 1.

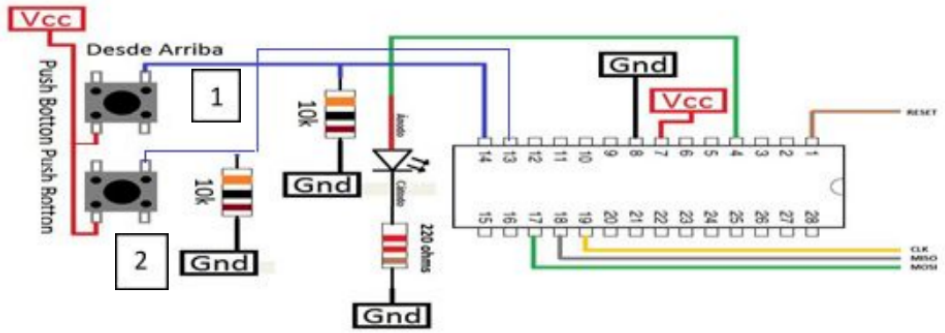

Figura 1: Esquemático utilizando una resistencia de *pull-up* externa.

Por último, se rediseñó la configuración del circuito y, consecuentemente, el programa desarrollado, al utilizar las resistencias de *pull-up* internas de los *ports* del ATMega328P.

3. Listado de componentes

A continuación, se detalla la lista de componentes utilizados, junto con los gastos que implicó el proyecto.

Material	Precio
Arduino UNO (placa imitación)	$835
Protoboard de 830 puntos	$277
Cables para *protoboard*	$119
2 pulsadores *Tact Switch*	$20
2 resistencias de 10 kΩ	$12
1 resistencia de 220 Ω	$6
1 diodo LED rojo	$15
Fletes	$550
TOTAL	$1834

Tabla 1: Tabla de gastos (en pesos argentinos).

4. Parpadeo de un diodo LED

Para la primera actividad a realizar se configuró el circuito representado en el esquemático de la figura 2. En el mismo, se utiliza una resistencia de 220 Ω, conectada al cátodo del diodo.

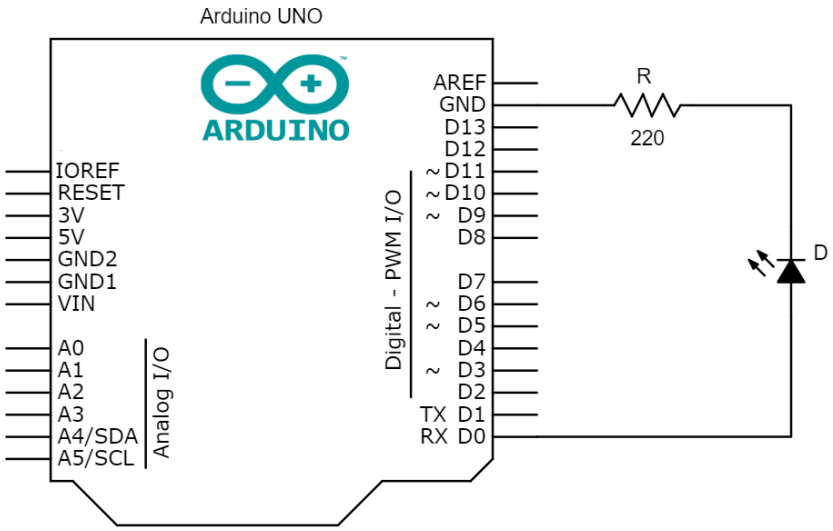

Figura 2: Esquemático para la primera práctica.

Al encender el LED, la salida del PIN 2 (puerto PD0) se encontrará en estado alto, por lo que en dicho terminal se establece una tensión de $V_{CC} = 5\,\text{V}$. Análogamente se da el caso opuesto, el estado bajo en 0 V del terminal de salida establecido en el ARDUINO implica que el diodo LED no emite luz. Dichas situaciones se muestran en las figuras 3 y 4.

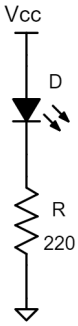

Figura 3: Representación para el caso con la salida D0 en estado alto (H).

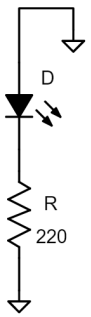

Figura 4: Representación para el caso con la salida D0 en estado bajo (L).

Utilizando las hojas de datos del diodo LED rojo difuso de 5 mm, se observó una tensión en directa típica de 2 V, junto con una corriente en directa máxima de 20 mA. De esta forma, tomando el valor de resistencia serie utilizada, la corriente que circula a través del diodo es

$$I_{LED} = \frac{V_{CC} - V_{LED}}{R} = \frac{5\,\text{V} - 2\,\text{V}}{220\,\Omega} = 13{,}6\,\text{mA}\,,$$

por lo que se encuentra por debajo de la intensidad máxima de corriente especificada.

4.1. Tiempo de retardo

Para lograr que se observen los efectos de la conmutación entre estados (encendido y apagado) del diodo, fue necesario diseñar una función que establezca un retardo entre instrucciones. Se

tomó arbitrariamente un tiempo de 250 ms, siendo el cristal de la plataforma de desarrollo de 16 MHz.

A partir de la rutina de retardo dada, se agregó un *loop* anidado adicional a fin de poder cumplir las condiciones impuestas. Cada ciclo de *clock*, en el cual se contabiliza la duración de cada instrucción, toma un valor de 62,5 ns. Luego, a partir del *delay* programado, se confeccionó la tabla 2, que indica los ciclos de *clock* que corresponden a cada instrucción, junto con sus repeticiones y el tiempo que se emplea en cada una.

Instrucción	Ciclos	Repeticiones	Duración
RCALL	3	1	0,19 µs
LDI TEMP1	1	1	62,5 ns
LDI TEMP2	1	243	15,2 µs
LDI TEMP3	1	$243 \cdot 229$	3,48 ms
DEC TEMP3	1	$243 \cdot 229 \cdot 23$	80 ms
BNEQ LOOP2 (T)	2	$243 \cdot 229 \cdot 22$	153 ms
BNEQ LOOP2 (F)	1	$243 \cdot 229$	3,48 ms
DEC TEMP2	1	$243 \cdot 229$	3,48 ms
BNEQ LOOP1 (T)	2	$243 \cdot 228$	6,93 ms
BNEQ LOOP1 (F)	1	243	15,2 µs
DEC TEMP1	1	243	15,2 µs
BNEQ LOOP0 (T)	2	242	30,3 µs
BNEQ LOOP0 (F)	1	1	62,5 ns
RET	4	1	250 ns
TOTAL		4007321	250,5 ms

Tabla 2: Tabla de ciclos, se muestra la cantidad total de instrucciones y el retardo que implican.

De esta forma, se obtiene un tiempo de retardo de aproximadamente 250 ms, en el que el LED se mantiene prendido antes de apagarse y viceversa.

4.2. Rutina de retardo

A continuación se muestra el código fuente de la rutina de retardo utilizada con los valores especificados en la tabla de la sección 4.1.

```
29 DELAY:              ; Retardo de 250ms a 16MHz
30 LDI T1, 243         ; Loops anidados
31 LOOP0:
32    LDI T2, 229
33 LOOP1:
34    LDI T3, 23
35 LOOP2:
36    DEC T3
37    BRNE LOOP2
38    DEC T2
39    BRNE LOOP1
40    DEC T1
41    BRNE LOOP0
42 RET
```

Se definieron los nombres de los registros `R20, R21` y `R22` a partir de la directiva `.DEF` como `T1, T2` y `T3`.

4.3. Circuito

En la figura 5 se muestra el circuito[1], conformado por un diodo LED y un resistor en serie de 220 Ω conectados entre dos *pines* del ARDUINO. Se utilizan los terminales GND y D0 (correspondiente al PIN 2 del microcontrolador).

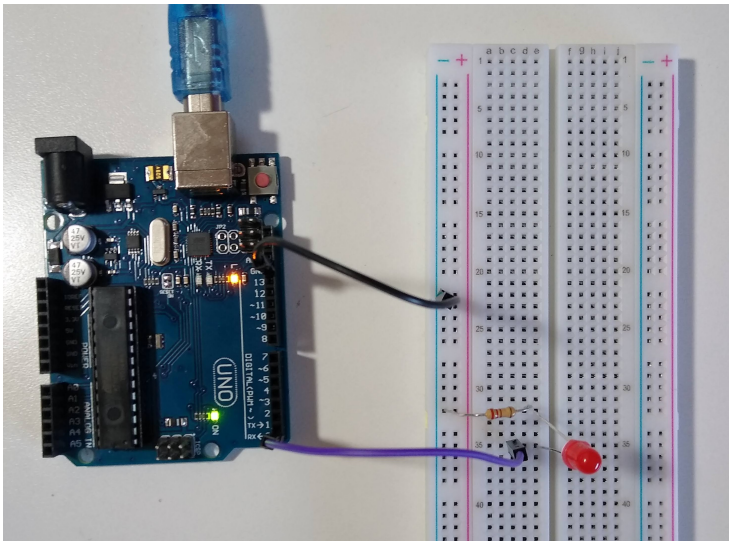

Figura 5: Circuito configurado para la primera práctica.

5. Prendido y apagado con pulsadores

Para la segunda experiencia práctica se configuró un circuito según el esquemático de la figura 6, donde se utilizan tres pines del microcontrolador. Como puertos de entrada se tienen los pines 13 y 14, correspondientes a los terminales D7 y D8 del ARDUINO, conectados a los pulsadores 2 y 1 respectivamente. Luego, el PIN 4, representado como D2, corresponde al puerto de salida que, dependiendo cuál de los pulsadores sea presionado, hará que el LED comience a parpadear o se apague.

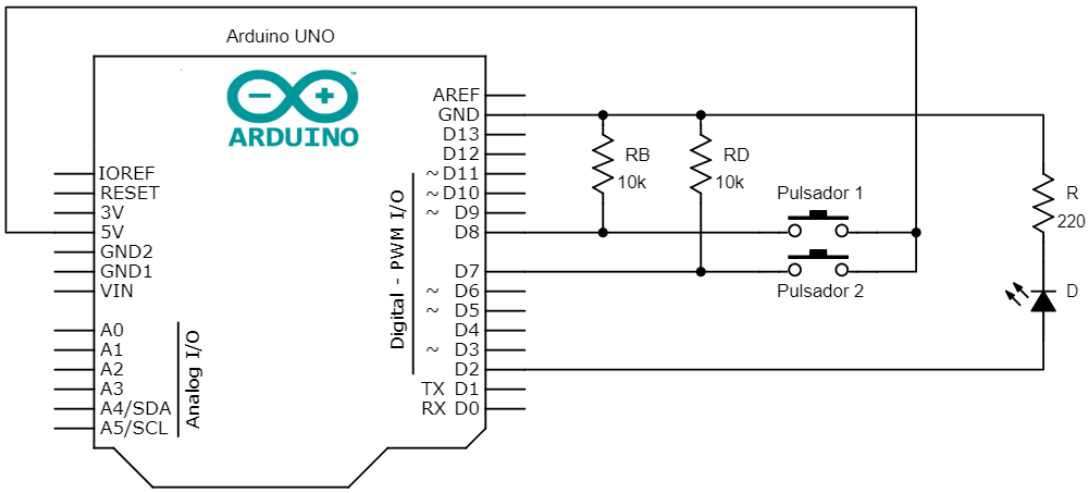

Figura 6: Esquemático para la segunda práctica.

[1]Link al video del circuito en funcionamiento.

5.1. Rutina de retardo

Para esta experiencia se modificó la rutina de retardo configurada inicialmente de modo que, al presionarse el pulsador 2, el LED se apague instantáneamente. Para esto, dentro de la rutina de retardo se lee la entrada D7, dado que el microcontrolador lee los valores instantáneos de los terminales.

A continuación se muestra la rutina implementada, que implica un tiempo de *delay* total de aproximadamente 500 ms.

```
83  ; Retardo de 500ms a 16MHz mientras se lee PD7
84  DELAY:
85  LDI T1, 243       ; Loops anidados
86  LOOP0:
87     LDI T2, 224
88  LOOP1:
89     LDI T3, 24
90  LOOP2:
91     IN RPIND, PIND ; Leo el registro PIN D
92     SBRC RPIND, 7  ; Si no se presiona el pulsador, no se llama a la rutina OFF
93     RJMP OFF
94     DEC T3
95     BRNE LOOP2
96     DEC T2
97     BRNE LOOP1
98     DEC T1
99     BRNE LOOP0
100 RET
```

Se definieron los nombres de los registros `R18` y `R19` como `RPINB` y `RPIND`, respectivamente, a modo de facilitar la legibilidad del código.

5.2. Rutina de apagado

En esta situación, se establece una rutina de apagado donde se impone un 0 lógico (L) en el puerto de salida del microcontrolador, para luego leer la entrada correspondiente al pulsador 1 y salir del *loop* una vez que este es presionado. A diferencia del caso anterior, no está definida la duración de la rutina programada.

```
74  ; Rutina que mantiene apagado el LED hasta que se presiona el pulsador 1
75  OFF:
76     CBI PORTD, 2   ; Apago el LED
77  READ:
78     IN RPINB, PINB ; Leo el registo PIN B
79     SBRC RPINB, 0  ; Si se apreta el pulsador 1, se salta al loop principal
80     RJMP LOOP
```

5.3. Diagrama de flujo

A continuación, en la figura 7, se expone un diagrama de flujo que representa la estructura básica del programa diseñado. El mismo se basa en un *loop* donde se leen las entradas conectadas a los pulsadores a cada paso que da el programa.

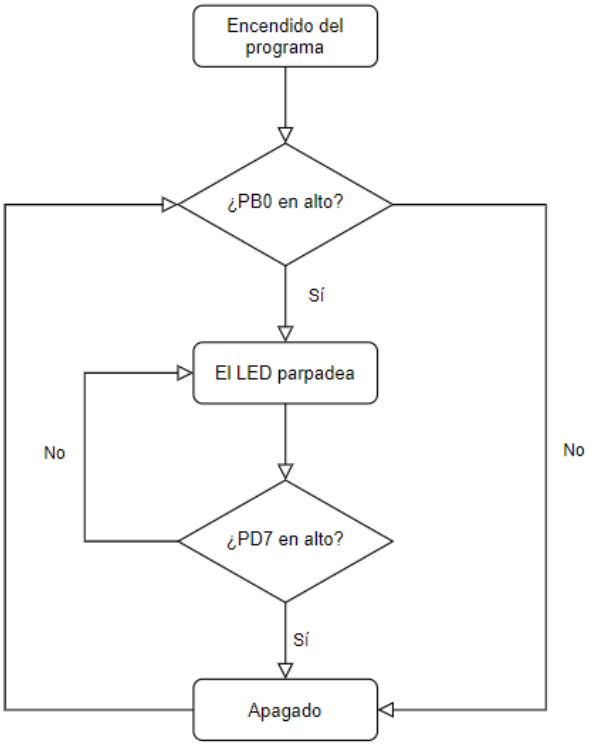

Figura 7: Diagrama de flujo representativo.

5.4. Circuito

En la figura 8 se muestra el circuito[2] con todas las conexiones establecidas. Para la alimentación se utiliza el PIN de 5 V del ARDUINO, que se observa en los esquemáticos.

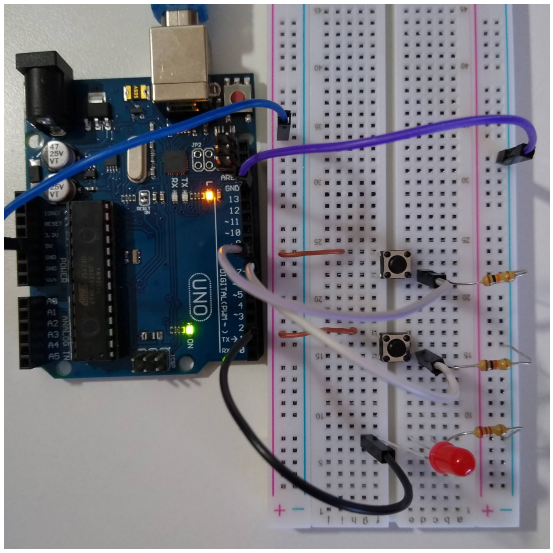

Figura 8: Circuito configurado para la segunda práctica.

6. Resistencias de *pull-up* internas

La configuración de *pull-up* implica que, al estar el circuito en reposo, es decir, sin presionar un pulsador, la caída de potencial en el pin correspondiente es de V_{CC} (H), en este caso 5 V,

[2]Link al video del circuito en funcionamiento.

mientras que cuando se presiona se impone un 0 lógico a la salida (L). La situación se representa en la figura 9, donde el nodo PIN indica la tensión que cae en un *pin* digital.

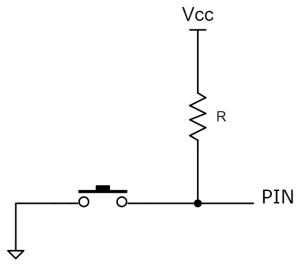

Figura 9: Configuración *pull-up*.

Luego, para aprovechar las resistencias internas del microcontrolador, se debió configurar el circuito de forma que los pulsadores estén conectados a tierra a fin de cerrar el circuito una vez apretado alguno de los pulsadores. De esta forma, se obtuvo el esquemático expuesto en la figura 10.

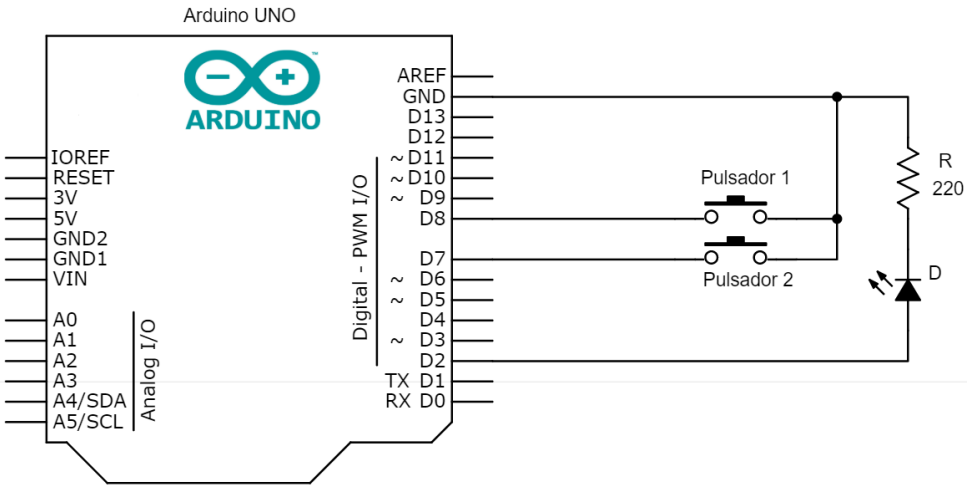

Figura 10: Esquemático para la tercera práctica.

En esta configuración se ahorró la colocación de los dos resistores de 10 kΩ conectados a los *pines*, junto con un cable que brinda los 5 V a los pulsadores.

6.1. Modificaciones de software

Dado que cambió la configuración del circuito, el programa debió ser modificado a fin de cumplir su funcionalidad.

En principio, se habilitaron las resistencias de *pull-up* para las dos entradas D7 y D8, escribiendo un 1 lógico en el *bit* correspondiente de cada registro. Además, en el programa se debieron intercambiar las instrucciones `SBRC` y `SBRS`, dado que, en este caso, la condición para una interrupción se da cuando en alguno de los *pines* se lee un estado bajo (L).

6.2. Circuito

En la figura 11 se muestra el circuito[3] conectado al ARDUINO. Se observa que, aprovechando la funcionalidad de las resistencias de *pull-up*, se tiene una menor cantidad de componentes y cables, lo que resulta en un circuito más compacto y ordenado.

[3]Link al video del circuito en funcionamiento.

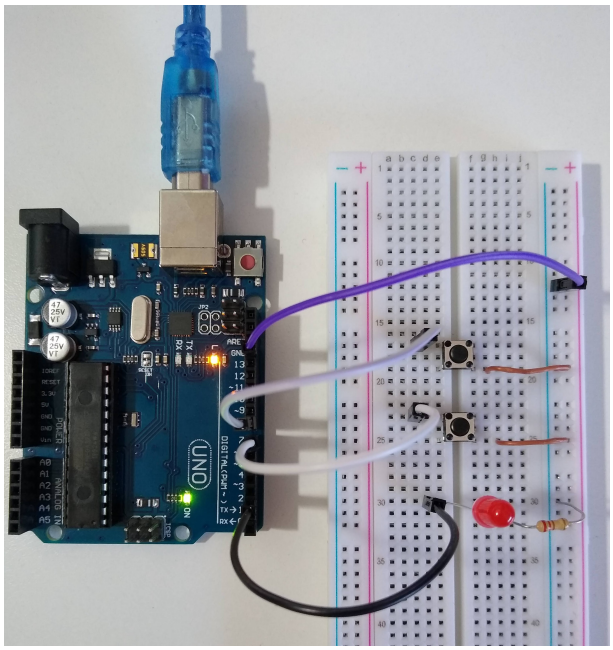

Figura 11: Circuito configurado para la tercera práctica.

7. Conclusiones

Este trabajo práctico logró introducir un primer acercamiento tanto al lenguaje de programación Assembly, como al manejo de puertos, funcionando como entradas y salidas. Se implementaron los conceptos adquiridos en el curso y se vieron reflejados los resultados esperados en cuando al comportamiento del sistema, el cual se desarrolló en una plataforma Arduino, ampliamente utilizada para distintos tipos de proyectos en el marco de la ingeniería.

Además se observó la utilidad de las resistencias de *pull-up* internas del microprocesador, dado que en circuitos más complejos y con un mayor número de componentes, es una herramienta que podría simplificar el armado del circuito físico.

8. Anexo

8.1. Esquemático completo del Arduino UNO

En la figura 12 se muestra el esquemático completo del Arduino utilizado. En el mismo se señalan en rojo las entradas utilizadas en las últimas dos experiencias, mientras que en azul se señala el puerto de salida del primer punto (PIN 2, D0) y el del segundo (PIN 4, D2). Además, en amarillo se muestran los terminales de 5 V y GND utilizados en las experiencias.

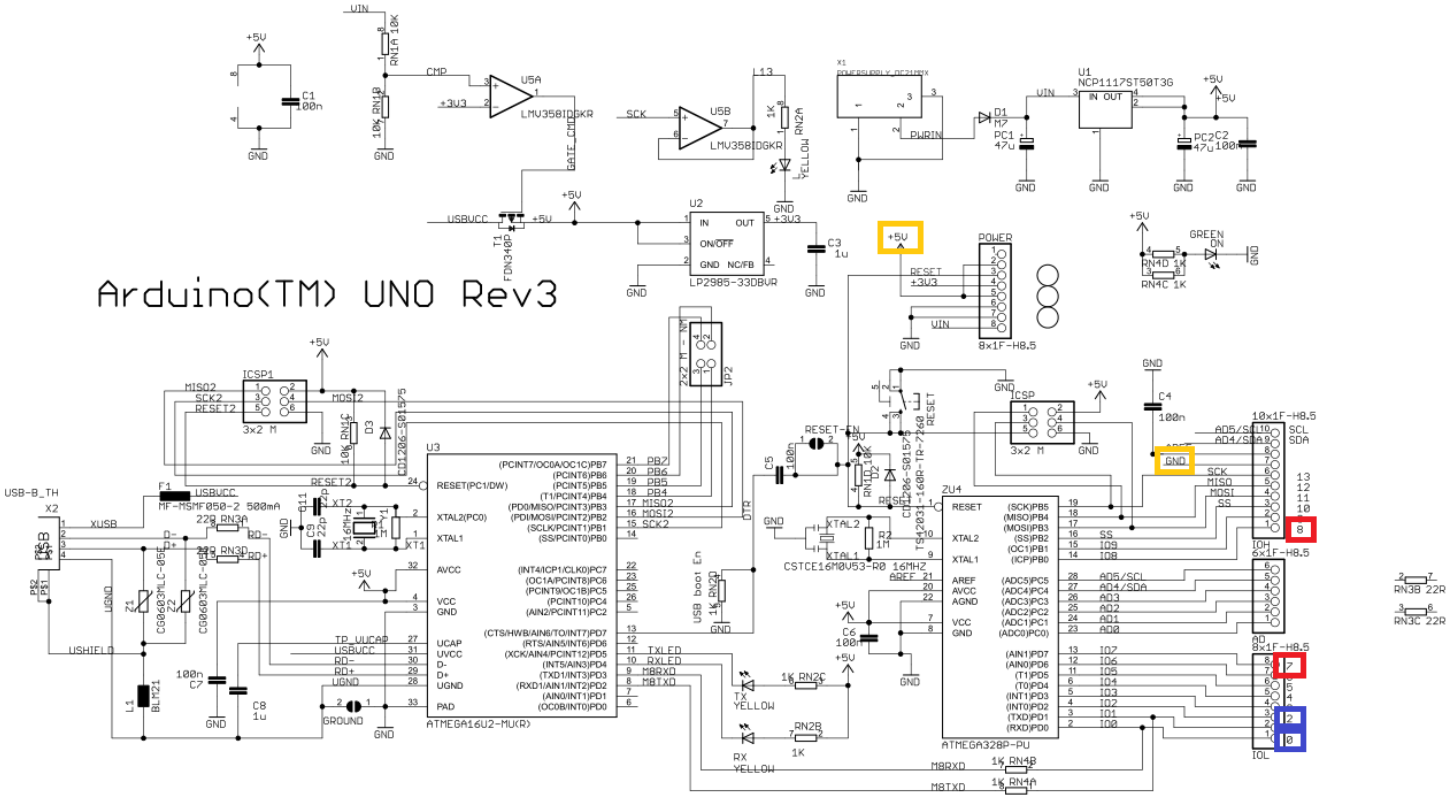

Figura 12: Esquemático del Arduino UNO utilizado.

8.2. Código fuente

```
; Diaz Falvo, Malena - 102374
; mcdiaz@fi.uba.ar
; 86.07 Laboratorio de Microprocesadores - 2C2020
; TP1
; ----------------------------------------------------

.DEF RPINB=R18
.DEF RPIND=R19
.DEF T1=R20
.DEF T2=R21
.DEF T3=R22
;.EQU BLINK=1
;.EQU BUTTONS=1
.EQU PULLUP=1


```

```
; Codigo para la primera practica
.IFDEF BLINK
LDI R16, 0B11111111
OUT DDRD, R16        ; Configuro el PORT D como un puerto de salida

LOOP:                ; Encendido y apagado del LED conectandolo al PIN 2 (PD0)
   SBI PORTD, 0      ; Prendo el LED
   RCALL DELAY       ; Impongo un retardo
   CBI PORTD, 0      ; Apago el LED
   RCALL DELAY       ; Vuelvo a imponer el retardo
   RJMP LOOP

DELAY:               ; Retardo de 250ms a 16MHz
LDI T1, 243          ; Loops anidados
LOOP0:
   LDI T2, 229
LOOP1:
   LDI T3, 23
LOOP2:
   DEC T3
   BRNE LOOP2
   DEC T2
   BRNE LOOP1
   DEC T1
   BRNE LOOP0
RET

.ENDIF


; Codigo para la segunda practica
.IFDEF BUTTONS

LDI R16, 0B00001111 ; Configuro la parte baja del PORT D como un puerto de
↪  salida
OUT DDRD, R16        ; y la alta como uno de entrada
LDI R16, 0B00000000
OUT DDRB, R16        ; Configuro el PORT B como un puerto de entrada

START:               ; Comienzo el programa con el LED apagado hasta que se
↪  presiona el pulsador 1
IN RPINB, PINB
SBRS RPINB, 0
RJMP START

LOOP:
   IN RPINB, PINB ; Leo el registro PIN B
   IN RPIND, PIND ; Leo el registro PIN D
   SBRC RPIND, 7  ; Si no se presiona el pulsador, no se llama a la rutina OFF
   RJMP OFF
   SBI PORTD, 2   ; Prendo el LED
   RCALL DELAY    ; Llamo a la rutina de retardo
```

```
67      SBRC RPIND, 7
68      RJMP OFF
69      CBI PORTD, 2    ; Apago el LED
70      RCALL DELAY
71      RJMP LOOP
72
73
74  ; Rutina que mantiene apagado el LED hasta que se presiona el pulsador 1
75  OFF:
76      CBI PORTD, 2    ; Apago el LED
77  READ:
78      IN RPINB, PINB ; Leo el registo PIN B
79      SBRC RPINB, 0  ; Si se apreta el pulsador 1, se salta al loop principal
80      RJMP LOOP
81      RJMP READ      ; Si no se apreta el pulsador 1, se sigue leyendo el PIN B
82
83  ; Retardo de 500ms a 16MHz mientras se lee PD7
84  DELAY:
85  LDI T1, 243        ; Loops anidados
86  LOOP0:
87      LDI T2, 224
88  LOOP1:
89      LDI T3, 24
90  LOOP2:
91      IN RPIND, PIND ; Leo el registro PIN D
92      SBRC RPIND, 7  ; Si no se presiona el pulsador, no se llama a la rutina OFF
93      RJMP OFF
94      DEC T3
95      BRNE LOOP2
96      DEC T2
97      BRNE LOOP1
98      DEC T1
99      BRNE LOOP0
100 RET
101
102 .ENDIF
103
104
105 ; Codigo para la tercera practica
106 .IFDEF PULLUP
107
108 LDI R16, 0B00001111  ; Parte baja del PORT D como un puerto de
109 OUT DDRD, R16        ; salida y la alta como uno de entrada
110 LDI R16, 0B00000000
111 OUT DDRB, R16        ; PORT B como un puerto de entrada
112 LDI R16, 0B00000001
113 OUT PORTB, R16       ; Resistencia de pull-up en el LSB
114 LDI R16, 0B10000000
115 OUT PORTD, R16       ; Resistencia de pull-up en el MSB
116
117 START:               ; Comienzo el programa con el LED apagado
118 IN RPINB, PINB       ; hasta que se presiona el pulsador 1
```

```
119 SBRC RPINB, 0
120 RJMP START
121
122 LOOP:
123    IN RPINB, PINB ; Leo el registro PIN B
124    IN RPIND, PIND ; Leo el registro PIN D
125    SBRS RPIND, 7  ; Si no se presiona el pulsador, no se llama a la rutina OFF
126    RJMP OFF
127    SBI PORTD, 2   ; Prendo el LED
128    RCALL DELAY    ; Llamo a la rutina de retardo
129    SBRS RPIND, 7
130    RJMP OFF
131    CBI PORTD, 2   ; Apago el LED
132    RCALL DELAY
133    RJMP LOOP
134
135
136 OFF:
137    CBI PORTD, 2   ; Apago el LED
138 READ:
139    IN RPINB, PINB ; Leo el registo PIN B
140    SBRS RPINB, 0  ; En caso de que se aprete el pulsador 1, se salta al loop
       ↪ principal
141    RJMP LOOP
142    RJMP READ      ; Si no se apreta el pulsador 1, se sigue leyendo el PIN B
143
144 DELAY:
145 LDI T1, 243
146 LOOP0:
147    LDI T2, 224
148 LOOP1:
149    LDI T3, 24
150 LOOP2:
151    IN RPIND, PIND ; Leo el registro PIN D
152    SBRS RPIND, 7  ; Si no se presiona el pulsador, no se llama a la rutina OFF
153    RJMP OFF
154    DEC T3
155    BRNE LOOP2
156    DEC T2
157    BRNE LOOP1
158    DEC T1
159    BRNE LOOP0
160 RET
161
162 .ENDIF
```