

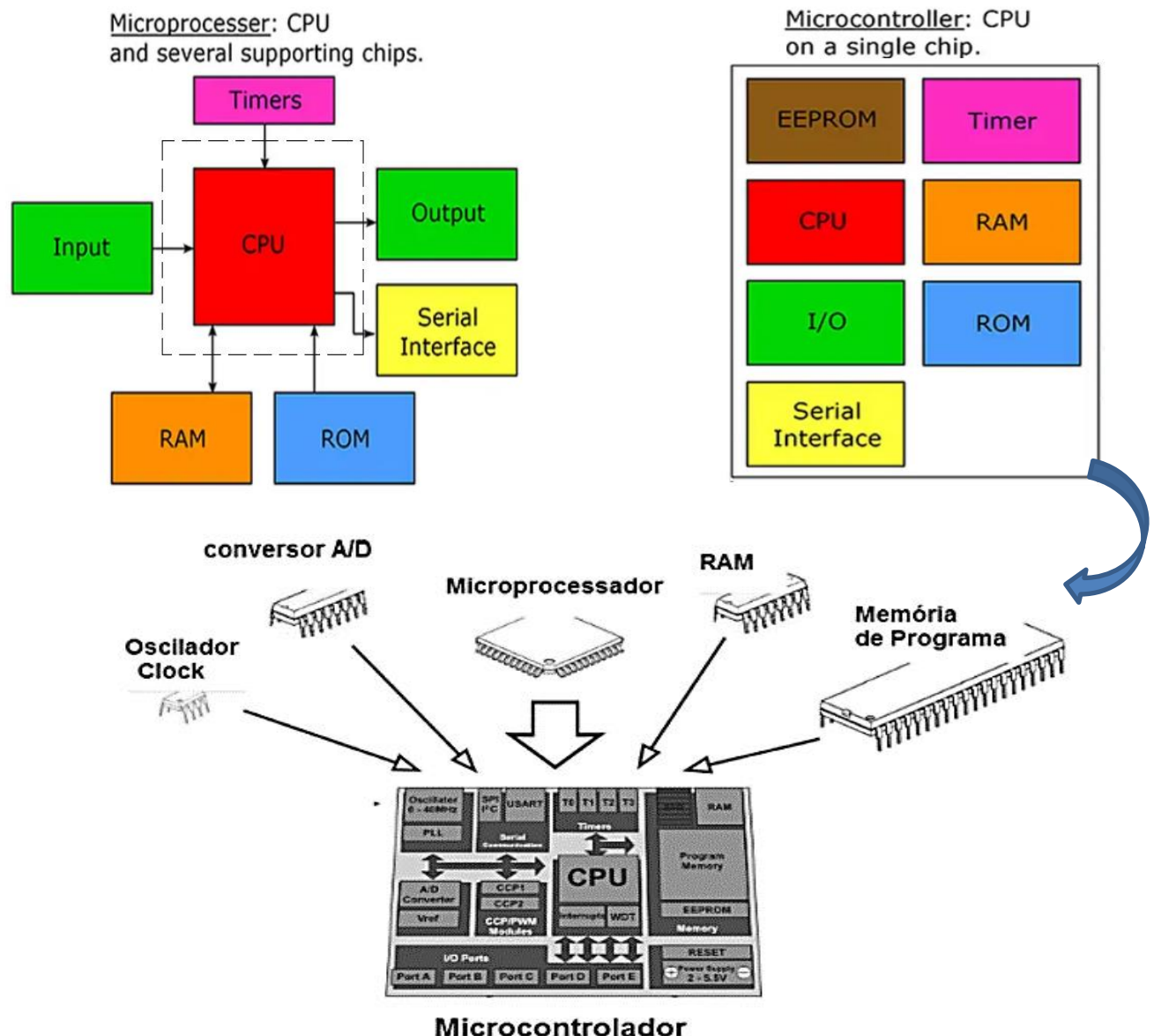
FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO
ASIGNATURA ELECTRONICA GENERAL Y APLICADA
CARRERA: INGENIERIA INDUSTRIAL. INGENIERIA MECATRONICA.
CICLO 2025 - CARPETA DE TRABAJOS PRACTICOS
TP N°8
TITULO: MICROCONTROLADOR. APLICACIONES

MICROCONTROLADOR vs. MICROCONTROLADOR

Un **Microprocesador** es la unidad de procesamiento central (CPU) de un sistema informático.

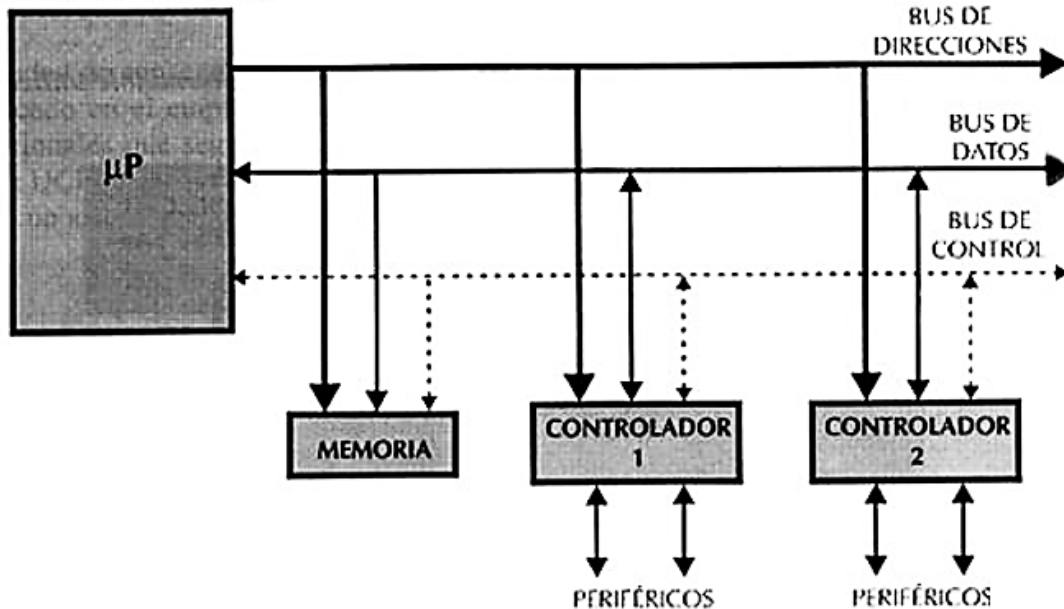
Un **Microcontrolador** es un sistema completo integrado en un solo chip, que contiene al microprocesador, memoria RAM y ROM, varios periféricos o interfaces de entrada/salida (E/S).

Los **microprocesadores** se usan en aplicaciones de alto rendimiento como computadoras, mientras que los **microcontroladores** son adecuados para tareas específicas en sistemas integrados muy variados.

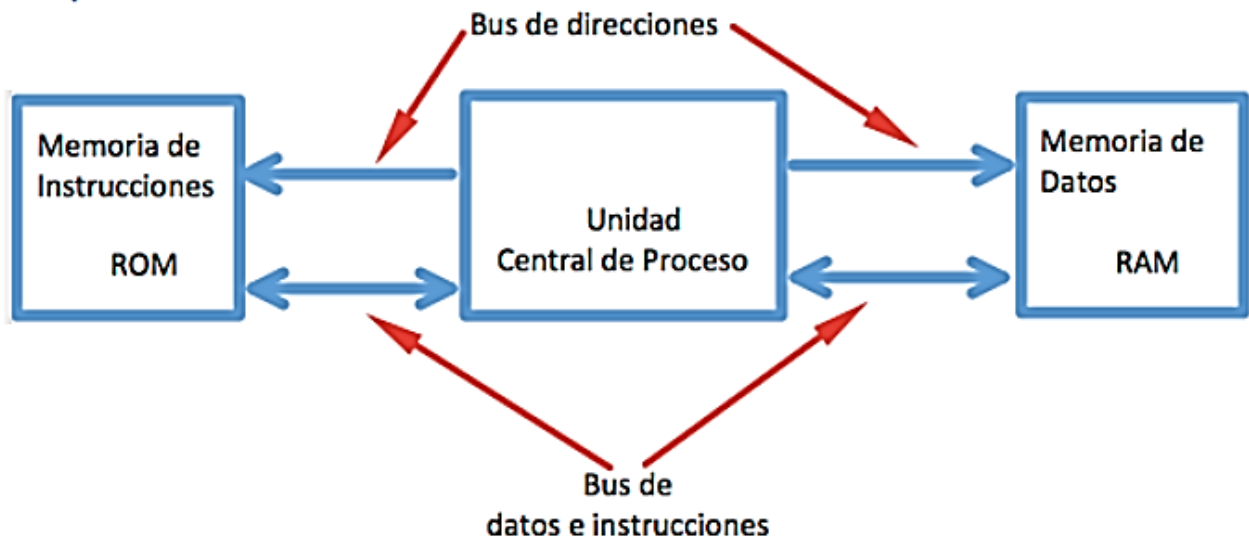


FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO
ASIGNATURA ELECTRONICA GENERAL Y APLICADA
CARRERA: INGENIERIA INDUSTRIAL. INGENIERIA MECATRONICA.
CICLO 2025 - CARPETA DE TRABAJOS PRACTICOS
TP N°8
TITULO: MICROCONTROLADOR. APLICACIONES

MICROPROCESADOR



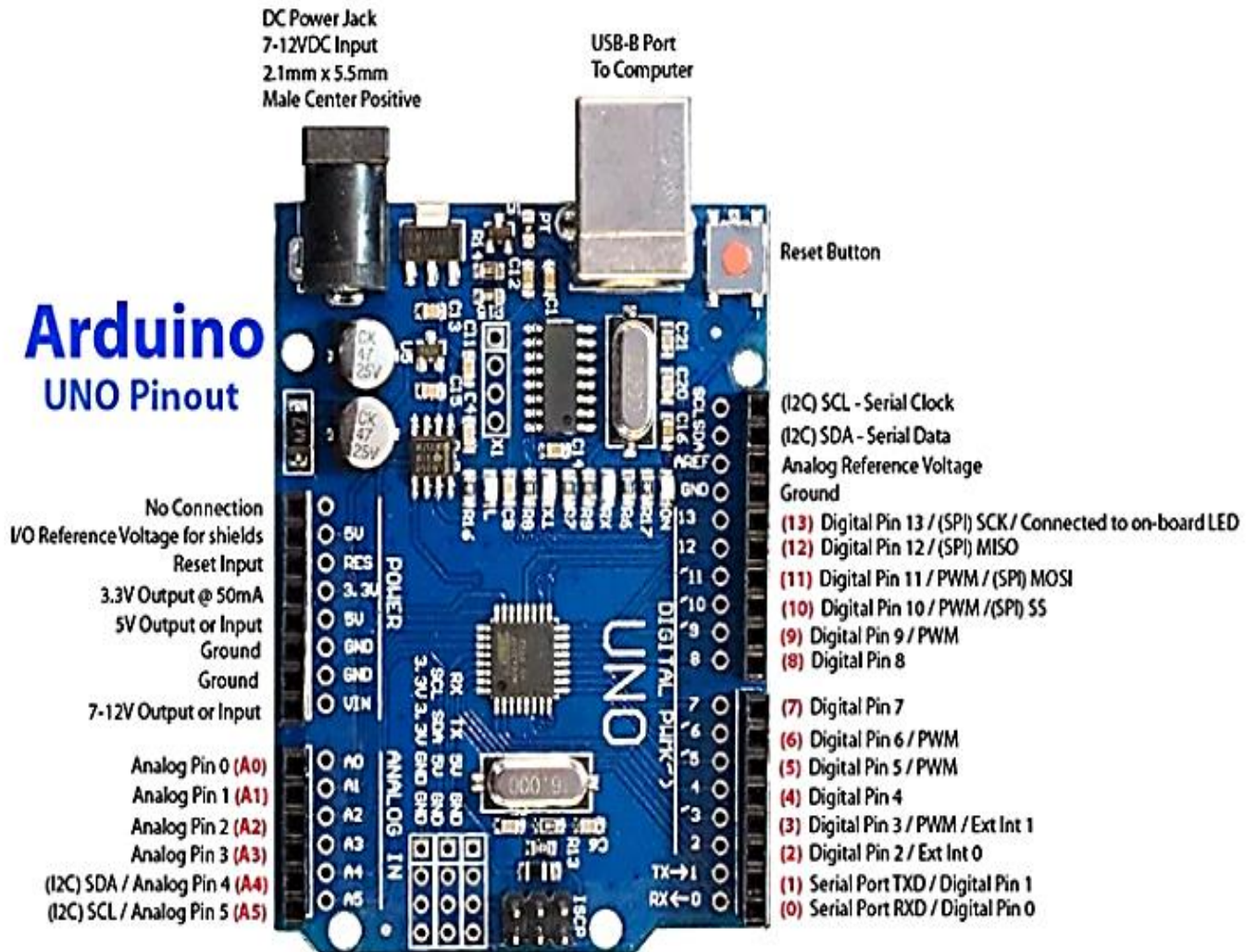
Arquitectura Harvard



Arquitectura Harvard: la memoria de programa y la memoria de datos están separadas, permitiendo el acceso simultáneo a ambos. Los microcontroladores en las placas Arduino, como el ATmega328, son los que implementan esta arquitectura, usando memoria flash para el código y SRAM para los datos.

FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO
ASIGNATURA ELECTRONICA GENERAL Y APLICADA
CARRERA: INGENIERIA INDUSTRIAL. INGENIERIA MECATRONICA.
CICLO 2025 - CARPETA DE TRABAJOS PRACTICOS
TP N°8
TITULO: MICROCONTROLADOR. APLICACIONES

MICROCONTROLADOR ARDUINO UNO



Red numbers in paranthesis are the name to use when referencing that pin.
Analog pins are references as A0 thru A5 even when using as digital I/O

FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO
ASIGNATURA ELECTRONICA GENERAL Y APLICADA
CARRERA: INGENIERIA INDUSTRIAL. INGENIERIA MECATRONICA.
CICLO 2025 - CARPETA DE TRABAJOS PRACTICOS
TP N°8
TITULO: MICROCONTROLADOR. APLICACIONES

ARDUINO. Resumen

Arduino es un microcontrolador de una placa de código abierto, que se utiliza en la construcción de proyectos de electrónica. Puede ser conectado a sensores, LEDs , motores, y otros dispositivos para crear una pantalla interactiva , kits de análisis , y cualquier otra cosa que la electrónica de base permita

Arduino ha sido popular entre los estudiantes, aficionados e incluso profesionales desde que se introdujo por primera vez en 2005. Antes de las placas basadas en Intel, la plataforma Arduino se compone de una cualquiera de 8 bits microcontrolador Atmel AVR Atmel ARM o una de 32 bits. Los últimos modelos de Arduino tienen 6 entradas analógicas y 14 pines de E / S digitales con una interfaz USB. Sin embargo, como con todas las tecnologías en desarrollo, las placas Arduino han evolucionado mucho.

Alimentación

El Arduino puede ser alimentado a través de la conexión USB o con un suministro de energía externo. La fuente de energía se selecciona mediante el jumper PWR_SEL: para alimentar la placa desde la conexión USB, colocarlo en los dos pines más cercanos al conector USB, para un suministro de energía externo, en los dos pines más cercanos al conector de alimentación externa. La alimentación externa (no USB) puede venir o desde un adaptador AC-a-DC o desde una batería. El adaptador puede ser conectado mediante un enchufe centro-positivo en el conector de alimentación de la placa. Los cables de la batería pueden insertarse en las cabeceras de los pines Gnd y Vin del conector POWER. Un regulador de tensión proporciona la tensión regulada al valor adecuado. Si se suministra menos de 7 Vcc, el pin de 5 V puede proporcionar menos de cinco voltios y la placa podría ser inestable. Si se usa más de 12 V, el regulador de tensión puede sobrecalentarse y dañar la placa. El rango recomendado es de 7 a 12 voltios.

Los pines de alimentación son los siguientes:

VIN. La entrada de tensión a la placa Arduino cuando está usando una fuente de alimentación externa (al contrario de los 5 voltios de la conexión USB u otra fuente de alimentación regulada). Puedes suministrar tensión a través de este pin, o, si suministra tensión a través del conector de alimentación, acceder a él a través de este pin.

5V. El suministro regulado de energía usado para alimentar al microcontrolador y otros componentes de la placa. Este puede venir o desde VIN a través de un regulador en la placa, o ser suministrado por USB u otro suministro regulado de 5 V.

3V3. Un suministro de 3.3 V generado por el chip FTDI de la placa. La corriente máxima es de 50 mA.

GND. Pines de Tierra.

FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO
ASIGNATURA ELECTRONICA GENERAL Y APLICADA
CARRERA: INGENIERIA INDUSTRIAL. INGENIERIA MECATRONICA.
CICLO 2025 - CARPETA DE TRABAJOS PRACTICOS
TP N°8
TITULO: MICROCONTROLADOR. APLICACIONES

Memoria

El ATmega168 tiene 16 KB de memoria Flash para almacenar código (de los cuales 2 KB se usa para el bootloader). Tiene 1 KB de SRAM y 512 bytes de EEPROM (que puede ser leída y escrita con la librería EEPROM).

Entradas y Salidas

Cada uno de los 14 pines digitales del UNO puede ser usado como entrada o salida, usando funciones `pinMode()`, `digitalWrite()` y `digitalRead()`. Operan a 5 voltios. Cada pin puede proporcionar o recibir un máximo de 40 mA y tiene una resistencia interna pull-up (desconectada por defecto) de 20-50 KOhms. Además, algunos pines tienen funciones especiales:

Serial: 0 (Rx) y 1 (Tx). Usados para recibir (Rx) y transmitir (Tx) datos TTL en serie. Estos pines están conectados a los pines correspondientes del chip FTDI USB-a-TTL Serie.

Interruptores externos: 2 y 3. Estos pines pueden ser configurados para disparar un interruptor en un valor bajo, un margen creciente o decreciente, o un cambio de valor.

PWM: 3, 5, 6, 9, 10 y 11. Proporcionan salida PWM de 8 bits con la función `analogWrite()`

SPI: 10 (SS), 11 (MOSI), 12 (MISO), 13 (SCK). Estos pines soportan comunicación SPI, la cual, aunque proporcionada por el hardware subyacente, no está actualmente incluida en el lenguaje Arduino.

LED: 13. Hay un LED empotrado conectado al pin digital 13. Cuando el pin está a valor HIGH, el LED está encendido, cuando el pin está a LOW, está apagado.

El UNO tiene 6 entradas analógicas, cada una de las cuales proporciona 10 bits de resolución (por ejemplo 1024 valores diferentes). Por defecto miden 5 voltios desde tierra, aunque es posible cambiar el valor más alto de su rango usando el pin AREF y algún código de bajo nivel. Además, algunos pines tienen funcionalidad especializada:

I²C: 4 (SDA) y 5 (SCL). Soportan comunicación I²C (TWI) usando la librería Wire5. Hay otro par de pines en la placa:

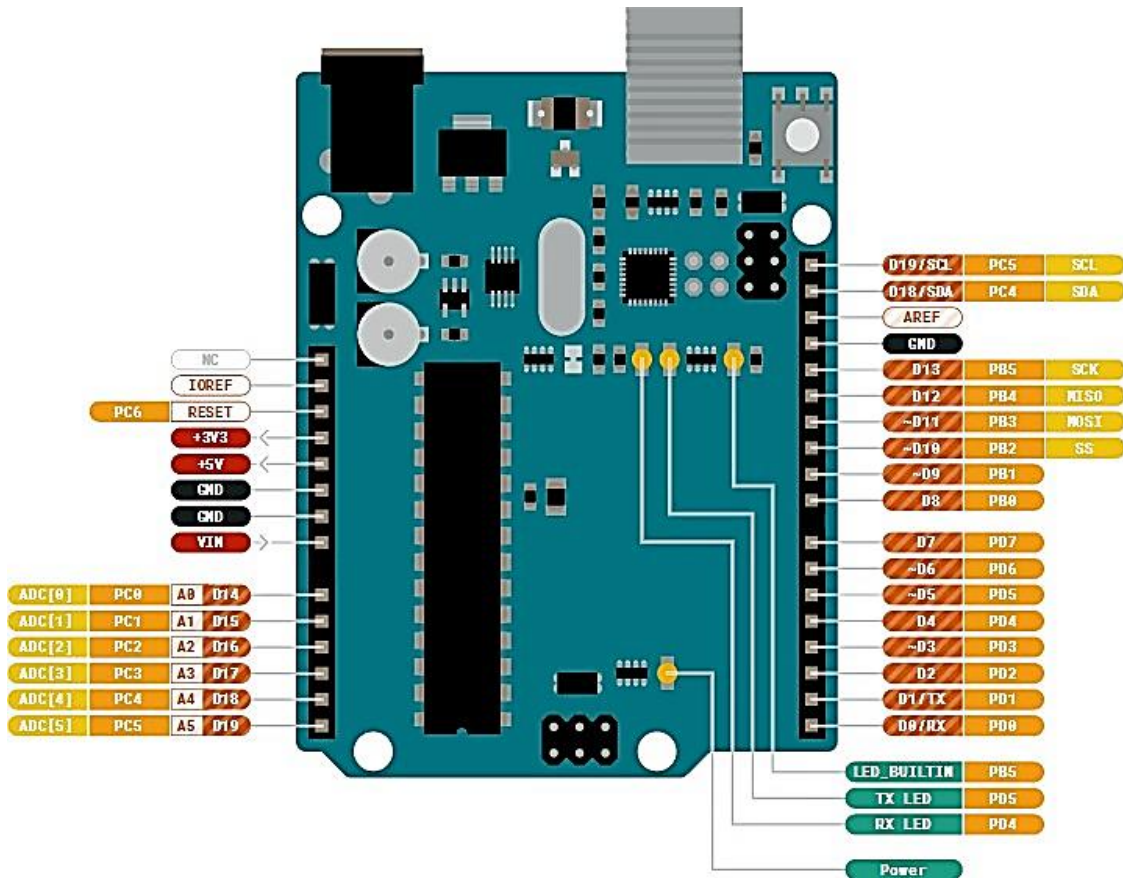
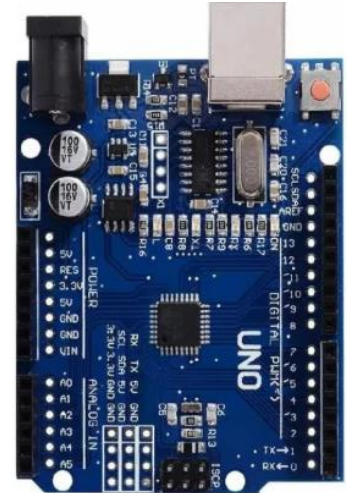
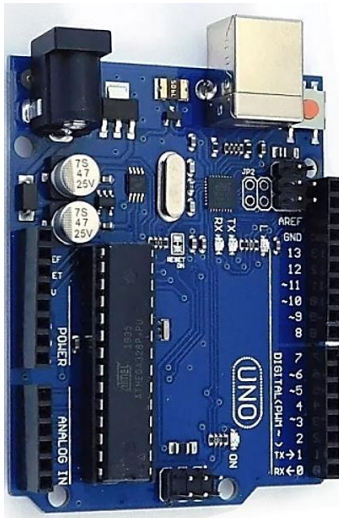
AREF. Voltaje de referencia para las entradas analógicas. Usado con `analogReference()`.

Reset. Pone esta línea a LOW para resetear el microcontrolador. Típicamente usada para añadir un botón de reset a dispositivos que bloquean a la placa principal.

TP N°8

TITULO: MICROCONTROLADOR. APLICACIONES

MICROCONTROLADOR ARDUINO UNO



FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO
ASIGNATURA ELECTRONICA GENERAL Y APLICADA
CARRERA: INGENIERIA INDUSTRIAL. INGENIERIA MECATRÓNICA.
CICLO 2025 - CARPETA DE TRABAJOS PRACTICOS
TP N°8
TITULO: MICROCONTROLADOR. APLICACIONES

APLICACION 1. MINIMIZACIÓN DE KARNAUGH

Un ejemplo a desarrollar es la posibilidad de realizar el circuito lógico de una Simplificación empleado el Mapa de Karnaugh (Trabajo Práctico 5) empleando Programación en el Microcontrolador.

¿Qué ventaja se obtiene? Con solo cambiar el Programa (software) Se logra modificar la aplicación sin necesidad de recurrir al armado de otro circuito cableado con compuertas lógicas (hardware).

Se toma como ejemplo el caso explicado

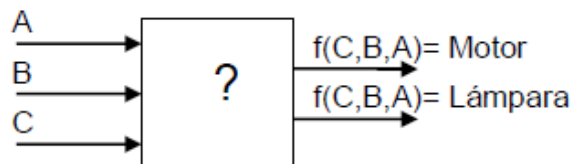
EJEMPLO 1

Un motor es controlado mediante tres pulsadores C,B,A. Diseñe un circuito de control mediante puertas lógicas que cumpla las siguientes condiciones de funcionamiento.

- Si se pulsan los tres pulsadores el motor se activa.
- Si se pulsan dos pulsadores cualesquiera el motor se activa pero se enciende una lámpara adicional como señal de emergencia.
- Si sólo se pulsa un pulsador, el motor no se activa, pero se activa la luz de emergencia.
- Si no se pulsa ningún pulsador, ni el motor ni la lámpara se activan.

Resolución:

Escribimos la tabla de la verdad para las dos funciones (motor y lámpara), luego minimizamos, y por último lo implementamos con compuertas.



FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO
ASIGNATURA ELECTRONICA GENERAL Y APLICADA
CARRERA: INGENIERIA INDUSTRIAL. INGENIERIA MECATRÓNICA.
CICLO 2025 - CARPETA DE TRABAJOS PRACTICOS
TP N°8
TITULO: MICROCONTROLADOR. APLICACIONES

TABLA DE VERDAD

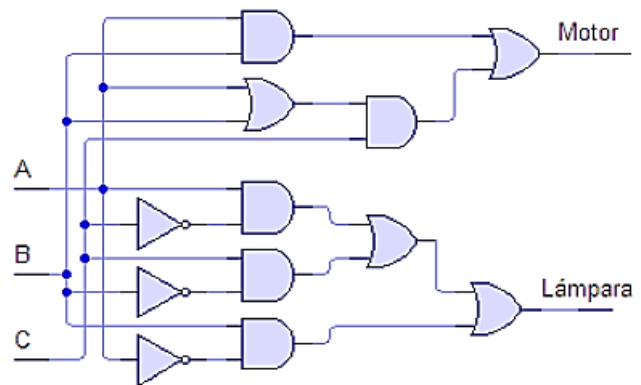
Nº	C	B	A	M	L
0	0	0	0	0	0
1	0	0	1	0	1
2	0	1	0	0	1
3	0	1	1	1	1
4	1	0	0	0	1
5	1	0	1	1	1
6	1	1	0	1	1
7	1	1	1	1	0

Circuito o esquema de la lógica binaria
(no tiene alimentación ni resistencias)

RESULTADO

$$\text{Motor} = f(C,B,A) = \text{CB} + \text{CA} + \text{BA}$$

$$\text{Lámpara} = f(C,B,A) = \bar{C}A + C\bar{B} + B\bar{A}$$



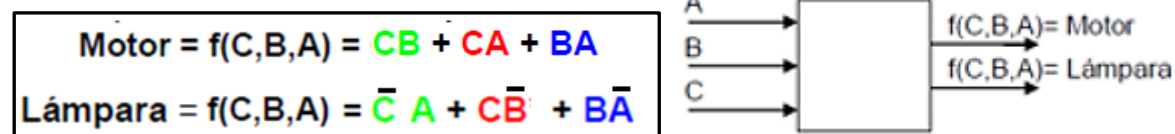
SOLUCIÓN

La solución es asociar las Entradas Lógicas a las Entradas Reales del Microcontrolador ARDUINO UNO y las Salidas Lógicas “M” y “L” (Motor y Lámpara) a las Salidas Reales del Arduino.

Para implementar la Aplicación se utiliza

- 1 Microcontrolador Arduino Uno
- 3 llaves de entradas asociadas a los pines 3,4 y 5 del Arduino UNO
- 1 Servomotor como Motor de la Aplicación
- 1 Led como Lámpara de la Aplicación.
- 1 Fuente de Alimentación: Se usa el puerto USB de la Notebook (5Vdc)
- 1 Programa implementado con el IDE ([Integrated Development Environment](#)) de Arduino
(es el software que permite desarrollar el programa y transferirlo al Microcontrolador)

Lógica que resuelve el problema



FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO
ASIGNATURA ELECTRONICA GENERAL Y APLICADA
CARRERA: INGENIERIA INDUSTRIAL. INGENIERIA MECATRONICA.
CICLO 2025 - CARPETA DE TRABAJOS PRACTICOS
TP N°8
TITULO: MICROCONTROLADOR. APLICACIONES

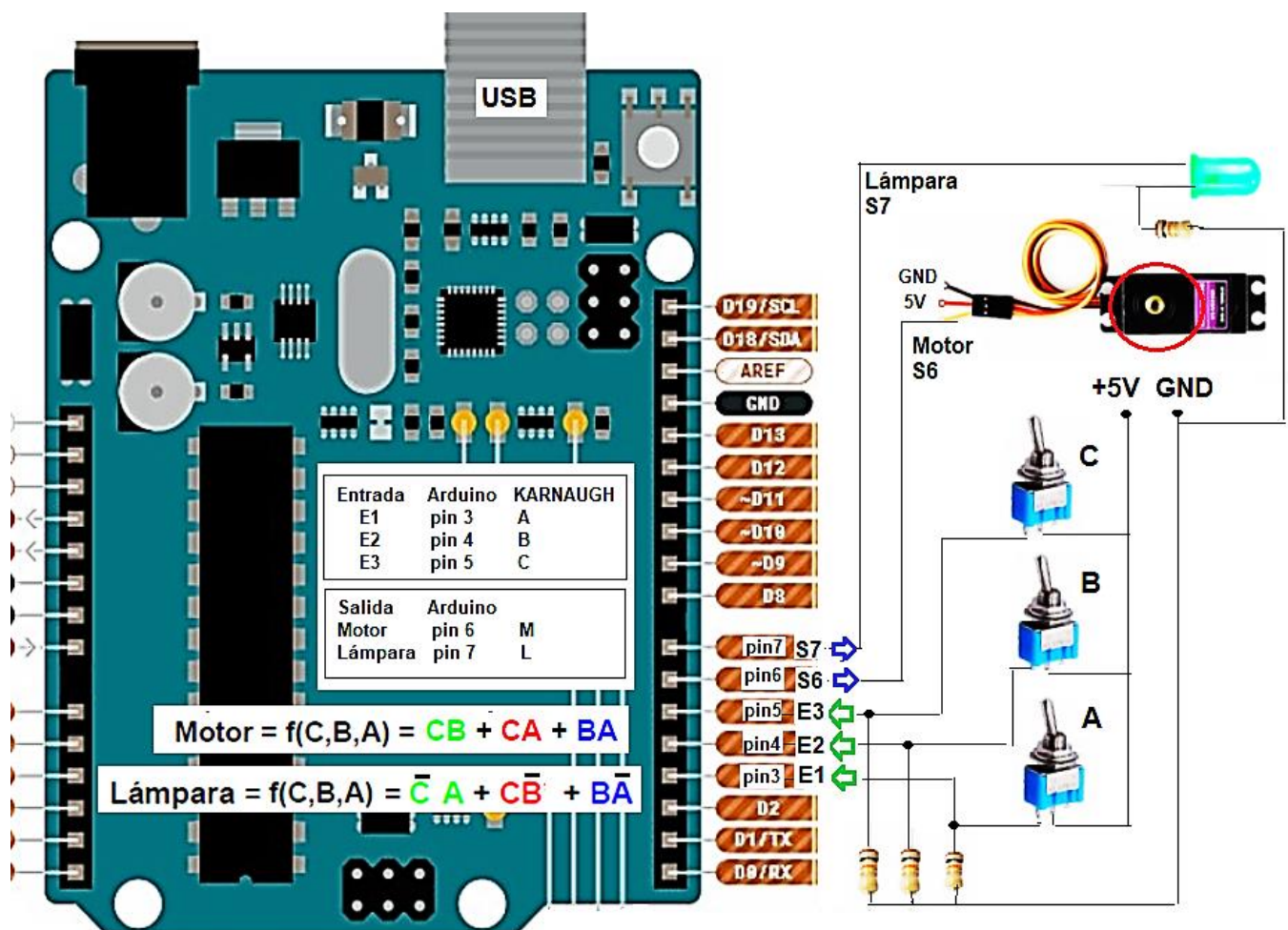
Para implementar el algoritmo (programa) se recurre a las instrucciones lógicas de Arduino

Compuerta	Instrucción
AND	&&
OR	
NOT	!

En el Algoritmo se incorporan las líneas de programa

```
motor= (entrada2 && entrada1) || (entrada3 && entrada1) || (entrada3 && entrada2);
lampara= (entrada3 && !entrada2) || (!entrada3 && entrada1) || (entrada2 && !entrada1);
```

Se asocia E1= entrada1, E2=entrada2, E3=entrada3. Esto es así porque se necesitan variables de programa (ej: entrada1) y variables (ej: E1) que asocian el pin real con el dato que entra o sale del microcontrolador.



FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO
ASIGNATURA ELECTRONICA GENERAL Y APLICADA
CARRERA: INGENIERIA INDUSTRIAL. INGENIERIA MECATRÓNICA.
CICLO 2025 - CARPETA DE TRABAJOS PRACTICOS
TP N°8
TITULO: MICROCONTROLADOR. APLICACIONES

CÓDIGO DEL PROGRAMA PARA ARDUINO

Se compone de 3 partes fundamentales: Inicio. Setup. Loop

INICIO

RUTINA_MOTOR_LAMPARA_MINIMIZADO.ino

```

1  // PROGRAMA CONTROL SISTEMA COMBINACIONAL SIMPLIFICADO METODO KARNAUHG
2  // 3 ENTRADAS ( A B C ) Y 2 SALIDAS M(Motor) y L (Lampara)
3  // TABLA DE VERDAD OBSERVADA EN TPN°5
4  // TPN°8 ELECTRONICA GRAL Y APLICADA.
5  // RUTINA SOBRE PLACA ARDUINO UNO. By ROBERTO HAARTH,02 Octubre 2025
6  // -----
7  #include <Servo.h>
8  Servo servo1; // Se declaran las variables del servomotor a utilizar.
9  //
10 // Se definen las variables asociadas a las señales uP (entradas/salidas)
11 #define E1 3 // ENTRADA A. conetado a pin3 uP.
12 #define E2 4 // ENTRADA B. conetado a pin3 uP.
13 #define E3 5 // ENTRADA C. conetado a pin3 uP.
14 #define M1 6 // SALIDA M motor conetado a pin6 uP. Motor gira o se detiene
15 #define L1 7 // SALIDA L lampara conetada a pin 7. Lampara que enciende o apaga
16
17 int entrada1 = 0; // Se almacena el estado o valor de ENTRADA A conetado al pin 3
18 int entrada2 = 0; // Se almacena el estado o valor de ENTRADA B conetado al pin 4
19 int entrada3 = 0; // Se almacena el estado o valor de ENTRADA C conetado al pin 5
20 int lampara1 = 0; // SALIDA LAMPARA
21 int motor1=0; // SALIDA SERVO 1
22 int motor=0; // VARIABLE KARNAUGH
23 int lampara=0; // VARIABLE KARNAUGH
24

```

FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO
ASIGNATURA ELECTRONICA GENERAL Y APLICADA
CARRERA: INGENIERIA INDUSTRIAL. INGENIERIA MECATRÓNICA.
CICLO 2025 - CARPETA DE TRABAJOS PRACTICOS
TP N°8
TITULO: MICROCONTROLADOR. APLICACIONES

SETUP

```

25 void setup()    {
26   Serial.begin(9600); // MONITOR SERIE ARDUINO 9600 NL&CR
27   servo1.attach(M1); // DEFINE SALIDA 6 PARA MOTOR M1
28   pinMode(E1, INPUT); // ENTRADA A
29   pinMode(E2, INPUT); // ENTRADA B
30   pinMode(E3, INPUT); // ENTRADA C
31   pinMode(L1, OUTPUT); // LAMPARA
32   delay(500); // introduce una Pausa en ms (solo es ejemplificativvo)
33   // EJECUTO ACCION DE SALIDA MOTOR DETENIDO Y LAMPARA APAGADA
34   digitalWrite(L1, LOW); // APAGA LAMPARA
35   servo1.write(112); // DETIENE MOTOR 1 EN LA SALIDA 6 (SERVO1)
36   delay(1000); // RETARDO DE 1 SEGUNDO
37   // EN ESTE ESPACIO (Setup) SE PUEDEN AGREGAR INSTRUCCIONES,
38   // PERO SOLO SE EJECUTAN UNA SOLA VEZ y SIGUE..
39   }
40

```

LOOP

```

41 void loop()    {
42   // INICIO DE LA RUTINA QUE ANALIZA LAS ENTRADAS Y EJECUTA LAS ACCIONES O COMANDOS
43   // ESTA EJECUCIÓN SE REPITE HASTA QUE ALGUNA ENTRADA O INTERRUPCION FINALIZA EL PROGRAMA.
44
45   entrada1=digitalRead(E1); //lee estado del pin 3 asignado a entrada A
46   entrada2=digitalRead(E2); //lee estado del pin 4 asignado a entrada B
47   entrada3=digitalRead(E3); //lee estado del pin 5 asignado a entrada C
48
49   // TABLA DE VERDAD A B C DISEÑO compuerta OR || NOT ! compuerta AND &&
50   // FUNCION MINIMIZADA DE KARNAUGH-----
51   motor= (entrada2 && entrada1) || (entrada3 && entrada1) || (entrada3 && entrada2);
52   lampara= (entrada3 && !entrada2) || (!entrada3 && entrada1) || (entrada2 && !entrada1);
53   // FUNCION MINIMIZADA DE KARNAUGH-----
54   // PREPARO SALIDA DE COMANDO DEL MOTOR
55   if(motor==LOW) // ANALIZA SALIDA FUNCION KARNAUGH PARA EL MOTOR
56   {
57     motor1=112; // VALOR DE COMANDO EN LA RUTINA SERVO QUE DETIENE EL MOTOR
58   }
59   else { // SIGNIFICA QUE motor1 TOMA EL VALOR HIGH
60     motor1=90; // VALOR DE COMANDO EN LA RUTINA SERVO QUE ENCIENDE EL MOTOR
61   }
62

```

FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO
ASIGNATURA ELECTRONICA GENERAL Y APLICADA
CARRERA: INGENIERIA INDUSTRIAL. INGENIERIA MECATRÓNICA.
CICLO 2025 - CARPETA DE TRABAJOS PRACTICOS
TP N°8
TITULO: MICROCONTROLADOR. APLICACIONES

```

63 // PREPARO SALIDA DE COMANDO DE LA LAMPARA
64 if(lampara==LOW) // ANALIZA SALIDA FUNCION KARNAUGH PARA EL MOTOR
65 |   |   | {
66 |   |   | lampara1=LOW;// VALOR DE COMANDO EN LA RUTINA SERVO QUE DETIENE EL MOTOR
67 |   |   | }
68 else { // SIGNIFICA QUE motor1 TOMA EL VALOR HIGH
69   lampara1=HIGH;// VALOR DE COMANDO EN LA RUTINA SERVO QUE ENCIENDE EL MOTOR
70 }
71 // EJECUTA SALIDAS DE COMANDO
72 |   |   | digitalWrite(L1, lampara1); // LAMAPARA EN LA SALIDA DIGITAL 7
73 |   |   | servo1.write(motor1);// MOTOR 1 EN LA SALIDA DIGITAL 6 (SERVO1)
74 //-----
75 |   |   | Serial.println("valores de A B C ");
76 |   |   | Serial.println(entrada1);
77 |   |   | Serial.println(entrada2);
78 |   |   | Serial.println(entrada3);
79 |   |   | Serial.println("motor");
80 |   |   | Serial.println(motor1);
81 |   |   | Serial.println("lampara");
82 |   |   | Serial.println(lampara1);
83 |   |   | Serial.println("-----");
84 delay (2000);// RETARDO DE TIEMPO. SOLO PARA MOSTRAR LOS ESTADOS TEMPORALES
85 }
86

```

FIN DEL PROGRAMA. **APLICACION 1.**

FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO
ASIGNATURA ELECTRONICA GENERAL Y APLICADA
CARRERA: INGENIERIA INDUSTRIAL. INGENIERIA MECATRÓNICA.
CICLO 2025 - CARPETA DE TRABAJOS PRACTICOS
TP N°8
TITULO: MICROCONTROLADOR. APLICACIONES

APLICACIÓN 2. CONTROL DE UN MONTACARGAS DE 3 PISOS.

Ejemplo 2. Control para un montacargas de 3 pisos

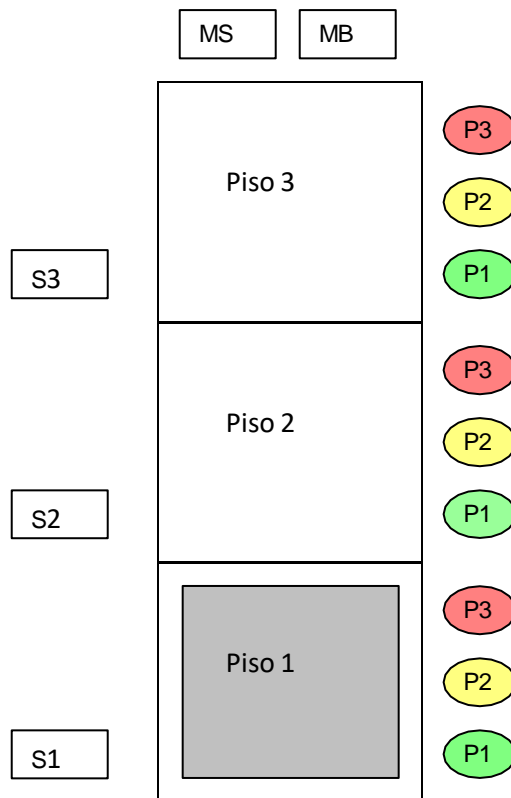
Como ejemplo de aplicación se realizará un programa para controlar un montacargas de 3 pisos.

El sistema tendrá las siguientes variables:

Entradas: 3 sensores de posición del montacargas, uno en cada piso. (Sp1, Sp2, Sp3)

3 pulsadores para llevar al montacargas al piso elegido. (Pp1, Pp2, Pp3)

Salidas : 2 una para cada sentido de giro del motor que moviliza al montacargas. (MS, MB)



Como primer paso realizaremos un diagrama de flujo que nos mostrara la secuencia que se deberá ejecutar para cumplir con el funcionamiento del elevador.

A partir del diagrama de flujo se podrá escribir el programa para cualquier dispositivo: microcomputador, P.L.C. etc., y para cualquier programa.

FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO
ASIGNATURA ELECTRONICA GENERAL Y APLICADA
CARRERA: INGENIERIA INDUSTRIAL. INGENIERIA MECATRONICA
CICLO 2025 - CARPETA DE TRABAJOS PRACTICOS
TP N°8
TITULO: MICROCONTROLADOR. APLICACIONES

CÓDIGO DEL PROGRAMA PARA ARDUINO

```

7  #define MS 10    // Motor subir
8  #define MB 8     // Motor bajar
9
10 #define S1 5     // sensor p1
11 #define S2 6     // sensor p2
12 #define S3 7     // sensor p3
13
14 #define P1 2     // pulsador p1  antes 2
15 #define P2 3     // pulsador p2
16 #define P3 4     // pulsador p3
17 int Pu1 = 0;
18 int Pu2 = 0;
19 int Pu3 = 0;
20 int Se1 = 0;
21 int Se2 = 0;
22 int Se3 = 0;
23 int Pisoactual = 0;
24
25 void setup()
26 {
27     pinMode( MB, OUTPUT); //MOTOR Bajar
28     pinMode( MS, OUTPUT); // MOTOR Subir
29     pinMode(S1, INPUT); //SENSOR P1
30     pinMode(S2, INPUT); //SENSOR P2
31     pinMode(S3, INPUT); //SENSOR P3
32     pinMode(P1, INPUT); //PULSADOR P1
33     pinMode(P2, INPUT); //PULSADOR P2
34     pinMode(P3, INPUT); //PULSADOR P3
35     Se1 = digitalRead(5); //lee el estado del pin 5 asignado al Sensor P1
36     if(Se1==LOW) //si no está en el P1
37     {
38         digitalWrite(MB,HIGH); // Motor bajar on
39     }
40     while (Se1==LOW)
41     {
42         Se1 =digitalRead(5);
43     }
44     digitalWrite(MB,LOW); // Motor bajar off
45     Pisoactual = 1; // Estamos en el piso 1
46
47 }
48

```

```

25 void setup()
26 {
27     pinMode( MB, OUTPUT); //MOTOR Bajar
28     pinMode( MS, OUTPUT); // MOTOR Subir
29     pinMode(S1, INPUT); //SENSOR P1
30     pinMode(S2, INPUT); //SENSOR P2
31     pinMode(S3, INPUT); //SENSOR P3
32     pinMode(P1, INPUT); //PULSADOR P1
33     pinMode(P2, INPUT); //PULSADOR P2
34     pinMode(P3, INPUT); //PULSADOR P3
35     Se1 = digitalRead(5); //lee el estado del pin 5 asignado al Sensor P1
36     if(Se1==LOW) //si no está en el P1
37     {
38         digitalWrite(MB,HIGH); // Motor bajar on
39     }
40     while (Se1==LOW)
41     {
42         Se1 =digitalRead(5);
43     }
44     digitalWrite(MB,LOW); // Motor bajar off
45     Pisoactual = 1; // Estamos en el piso 1
46
47 }
48
49 void loop()
50 {
51
52
53     Pu1=digitalRead(P1); //lee el estado del pin 2 asignado al pulsador del P1
54     Pu2=digitalRead(P2); //lee el estado del pin 3 asignado al pulsador del P2
55     Pu3=digitalRead(P3); //lee el estado del pin 4 asignado al pulsador del P3
56
57     if(Pu1==HIGH) //si se pulsa el pulsador del P1
58     {
59         if(Pisoactual==2) //
60         {
61             digitalWrite(MB, HIGH); // Motor bajar
62
63             while (Se1==LOW)
64             {
65                 Se1 =digitalRead(S1);
66             }
67
68             digitalWrite(MB,LOW);
69             Pisoactual = 1; // Estamos en el piso 1
70         }
71
72     else if(Pisoactual==3) //si esta la cabina en la planta tercera el ascensor baja mas tiempo
73     {
74         digitalWrite(MB, HIGH); // Activamos Motor
75

```

```
76 while (Se1==LOW)
77 {
78   Se1 =digitalRead(S1);
79 }
80
81   digitalWrite(MB,LOW);
82   Pisoactual = 1; // Estamos en el piso 1
83 }
84 | }
85
86 else if(Pu2==HIGH) //
87 {
88   if(Pisoactual==1) //y la planta actual es la primera
89   {
90     digitalWrite(MS, HIGH);      // Activamos Motor
91
92     while (Se2==LOW)
93     {
94       Se2 =digitalRead(S2);
95
96     }
97
98     digitalWrite(MS,LOW);
99     Pisoactual = 2; // Estamos en el piso 2
100   }
101   else if (Pisoactual==3)
102   {
103     digitalWrite(MB, HIGH);      // Activamos Motor
104
105     while (Se2==LOW)
106     {
107       Se2 =digitalRead(S2);
108     }
109
110     digitalWrite(MB,LOW);
111     Pisoactual = 2;
112   }
113 }
114 else if(Pu3==HIGH)
115 {
116   if(Pisoactual==1)
117   {
118     digitalWrite(MS, HIGH);
119
120     while (Se3==LOW)
121     {
122       Se3 =digitalRead(S3);
123     }
124
```

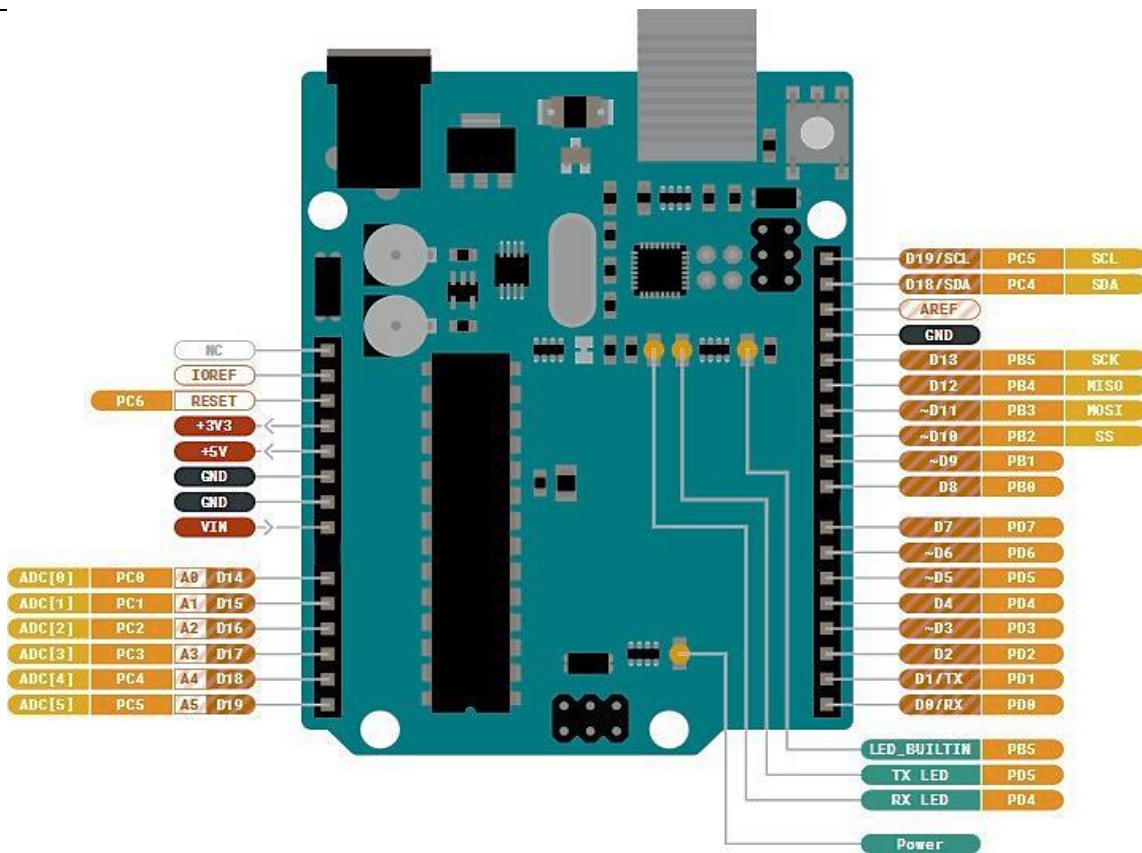
FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO
ASIGNATURA ELECTRONICA GENERAL Y APLICADA
CARRERA: INGENIERIA INDUSTRIAL. INGENIERIA MECATRONICA
CICLO 2025 - CARPETA DE TRABAJOS PRACTICOS
TP N°8
TITULO: MICROCONTROLADOR. APLICACIONES

```

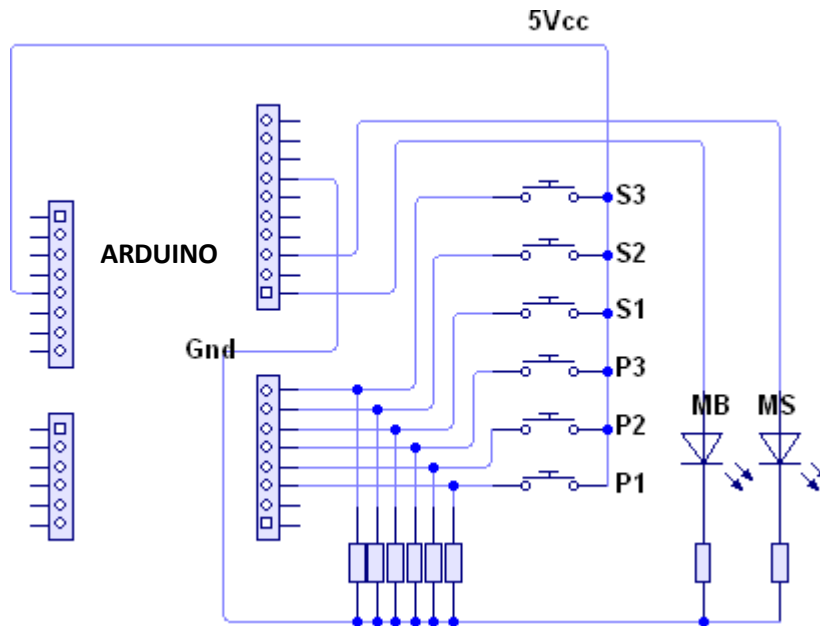
125         digitalWrite(MS,LOW);
126         Pisoactual = 3;
127     }
128     else if (Pisoactual==2)
129     {
130         digitalWrite(MS, HIGH);
131
132         while (Se3==LOW)
133         {
134             Se3 =digitalRead(S3);
135         }
136
137         digitalWrite(MS,LOW);
138         Pisoactual = 3; //
139     }
140 }
141 Se1 = 0;
142 Se2 = 0;
143 Se3 = 0;
144 }
145

```

FIN DEL PROGRAMA. APLICACIÓN 2.



CABLEADO



FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO
ASIGNATURA ELECTRONICA GENERAL Y APLICADA
CARRERA: INGENIERIA INDUSTRIAL. INGENIERIA MECATRONICA
CICLO 2025 - CARPETA DE TRABAJOS PRACTICOS
TP N°8
TITULO: MICROCONTROLADOR. APLICACIONES

TRABAJO PRACTICO A DESARROLLAR

Punto A

Responda las siguientes preguntas

- a) En el siguiente fragmento de código en qué piso queda el elevador si P1 = 0? `if(Pu1==HIGH)`

```

{
  if(Pisoactual==2)
  {
    digitalWrite(MB, HIGH);

    while (Se1==LOW)
    {
      Se1 =digitalRead(S1);
    }

    digitalWrite(MB,LOW);
    Pisoactual = 1;
  }

```

- b) En el siguiente fragmento de código qué debe ocurrir para que salga del bucle resaltado el rojo?

```

if(Pu1==HIGH)
{
  if(Pisoactual==2)
  {
    digitalWrite(MB, HIGH);

    while (Se2==LOW)
    {
      Se2 =digitalRead(S2);
    }

    digitalWrite(MB,LOW);
    Pisoactual = 1;
  }

```

FACULTAD DE INGENIERIA –UNIVERSIDAD NACIONAL DE CUYO
ASIGNATURA ELECTRONICA GENERAL Y APLICADA
CARRERA: INGENIERIA INDUSTRIAL. INGENIERIA MECATRONICA
CICLO 2025 - CARPETA DE TRABAJOS PRACTICOS
TP N°8
TITULO: MICROCONTROLADOR. APLICACIONES

Punto B.

c) En el siguiente fragmento de código explique que se realiza

```
motor= (entrada2 && entrada1) || (entrada3 && entrada1) || (entrada3 && entrada2);
lampara= (entrada3 && !entrada2) || (!entrada3 && entrada1) || (entrada2 && !entrada1);
// FUNCION MINIMIZADA DE KARNAUGH-----
// PREPARO SALIDA DE COMANDO DEL MOTOR
| | | | if(motor==LOW) // ANALIZA SALIDA FUNCION KARNAUGH PARA EL MOTOR
| | | | {
| | | | | motor1=112;// VALOR DE COMANDO EN LA RUTINA SERVO QUE DETIENE EL MOTOR
| | | | }
| else { // SIGNIFICA QUE motor1 TOMA EL VALOR HIGH
| | | | motor1=90;// VALOR DE COMANDO EN LA RUTINA SERVO QUE ENCIENDE EL MOTOR
| | | | }
```

d) Escriba la función lógica para la función minimizada

$$\text{Motor} = f(C,B,A) = C(B+A) + BA$$

Nota. Utilice las mismas variables del programa, esto es

Variables entrada1 para "A". entrada2 para "B", entrada3 para "C"