

## **1. DESCRIPCIÓN DEL PROBLEMA**

La institución universitaria Pascual Bravo líder en Educación Superior Tecnológica, comprometida socialmente con la formación de profesionales íntegros, con certificación de calidad en procesos y en busca de la excelencia académica, a través de modelos pedagógicos dinámicos que respondan a las necesidades de la región y del país.

En la institución no se han hecho trabajo relacionados con este proyecto; la institución tiene varias pantallas de leds en el CIS y en la biblioteca; pero estas fueron compradas para la información de turnos u horarios.

La institución en vez de comprar por fuera estos tableros podría utilizar estos recursos para el patrocinio de proyectos dentro de la institución y así fomentar la investigación en el estudiante.

Los tableros publik que están instalados en la institución tienen una tecnología muy simple y pueden tener muchas falencias en su diseño, debido a que sus diseñadores no dejan hacer cambios ni actualizaciones. al no tener acceso a la información de estos tableros en su software ni en hardware a corto plazo se pueden volver desechables por posibles daños, ya que de estos no se tiene la información para poderlos modificar o arreglar.

### **1.1 FORMULACIÓN DEL PROBLEMA**

¿Se podrán cambiar los carteles de la entrada por estos avisos para el pico y placa?

## 2. JUSTIFICACIÓN

En el proyecto se van a poner a prueba los conocimientos y las capacidades que se han adquirido en la especialidad (Desarrollo y mantenimiento de equipo) durante todos estos años que llevamos en la institución.

Este proyecto se va a elaborar porque es algo muy innovador y de gran utilidad ya que se puede utilizar en muchos casos y situaciones, la utilización de este puede ser tanto en interiores como en exteriores, este enfatizará en interiores porque es más acorde, ya que el lugar de exposición debe de ser un salón.

Además durante la elaboración del proyecto a lo largo de estos meses se van a conocer las capacidades que se han adquirido durante estos años en la institución, además para el proyecto se necesita investigar e indagar con más personas las dudas que puedan surgir en su elaboración, para así lograr un conocimiento más fuerte y estructurado el tema estudiado.

Así se tendrá el propósito de una buena publicidad e información dentro de la institución. El beneficio que se recibirá en la institución es muy grande ya que a través de este prototipo la comunidad estudiantil podría enterada de todos los sucesos que están pasando dentro de la institución.

La riqueza técnica del proyecto es su bajo consumo de energía y por la utilización de materiales ecológicos como son los leds y su tecnología nueva por su micro controlador arduino.

### **3. OBJETIVOS**

#### **3.1 OBJETIVO GENERAL**

Diseñar un tablero publik que permita visualizar información de la institución universitaria Pascual Bravo.

#### **3.2 OBJETIVOS ESPECÍFICOS**

Integrar los conocimientos teóricos con el diseño y programación de un tablero publik.

Programar los tableros con el software apropiado que permita la utilización óptima del tablero publik, integrar los componentes al circuito que permita visualizar la programación.

Realizar pruebas de funcionamiento de tablero publik.

Entregar un prototipo de tablero publik para que sirva como soporte para la elaboración de otros tableros a gran escala.

#### 4. MARCO TEÓRICO

Llamado “Publik” en algunos países, es un recurso utilizado muy frecuentemente con fines publicitarios o informativos, básicamente, consiste en una matriz de pixeles, generalmente de un solo color (rojos), aunque con el descenso de los precios de los LEDs individuales o en paneles, es cada vez más frecuentes ver carteles “bicolores” o incluso multicolores, aprovechando la ventaja de los LEDs RGB, que pueden mostrar cualquier color.

Por motivos de simplificar el circuito y de no gastar demasiado dinero, nuestro cartel será monocromático, utilizada serán de 7 filas por 30 columnas, lo que permite escribir unas 14 ó 16 letras de 7 “pixeles” de altura, a pesar de no ser demasiado grande, ya habrás sacado la cuenta de que se necesitan 210 LEDs para armar el cartel.

El secreto está en el multiplexado, es decir, utilizar unos pocos pines de E/S del microcontrolador para manejar una serie de circuitos integrados que se encarguen de excitar las LEDs.

Ya mencionado anteriormente, la pantalla está formada por una serie de filas y columnas, la intersección entre ambas contiene un LED. Para que este encienda, tiene que recibir simultáneamente un “0” en la fila, y un “1” en las columnas, cuando se dan estas condiciones, la electrónica de la placa se encarga del encendido. La forma de generar un mensaje sobre el display es relativamente sencilla, si nos atenemos al siguiente algoritmo:

Apagar todas las filas.

Escribir los valores correspondientes a la primer fila en el registro de desplazamiento, teniendo en cuenta que el primer dígito binario colocado corresponde al último LED de la fila, y el último en poner al de la primera columna.

Encender la primera fila, esperar un tiempo, volver a apagar y repetir los pasos para las filas 2 a 7.

Los tiempos de demora que se utilizaran en el programa de ejemplo permiten una visualización correcta, sin molestos parpadeos y con los LEDs brillantes. Hay que tener en cuenta que si se utilizan tiempos mayores para el encendido de cada fila, el brillo de los LEDs será mayor, pero también aumenta el parpadeo. No se utilizarán vectores, otras alternativas que hubieran servido para crear un código más compacto, buscando la claridad del programa, para que pueda servir como base a otros más completos/complejos.

Un punto a tener en cuenta es el brillo de los LEDs. Un LED, utilizado en aplicaciones “normalmente”, se alimenta con unos 3V y requiere unos 15mA para brillar con una buena intensidad. En este caso, a pesar de que se verán las 7 filas encendidas al mismo tiempo, cada LED solo estará encendido la séptima parte del tiempo, por lo que su brillo será siete veces inferior al normal, y nuestro cartel apenas será visible. Afortunadamente eso tiene solución: dado que los tiempos que permanecerá encendido cada LED no supera unos pocos milisegundos, no muchos más intensamente, dando como resultado un cartel perfectamente visible.

#### **4.1 EL CIRCUITO CONTROLADOR**

Será el encargado de gestionar el encendido de cada LED mediante órdenes enviadas a las columnas (mediante el registro de desplazamientos que mociónamos en la nota anterior) y a las filas.

No se incluirá o una fuente de alimentación, cualquier fuente comercial que sea capaz de entregar 5 V y 2 A será suficiente. Esos 5V deben estar bien regulados, y por supuesto, el software deberá estar escrito correctamente, es decir, no encender varias filas al mismo tiempo, ya que el consumo de todo el cartel encendido sería de unos  $80 \times 7020 \text{mA} = 11.2 \text{ A}$ , destruyendo la fuente.

## **4.2 EL SOFTWARE DEL MICROCONTROLADOR**

El software, es la información codificada, que es transmitida al hardware, para que este la procese y la ejecute. El hardware, son todos los elementos físicos, que componen la computadora, por lo mismo, estos necesitan del software, para funcionar.

En otras palabras, el software son los programas que utilizan las computadoras para funcionar, en cuanto a la utilización del software en computadoras, este debe ser cargado en espólon o memoria interna, el cual es ejecutado en la unidad central del mismo o CPU.

El lenguaje utilizado por el software, para comunicarse con el hardware, es de tipo binario, el cual sólo es ocupado por elementos electrónicos o tecnológicos. Pero todo este lenguaje viene en forma de instrucciones, las cuales son ejecutadas, por cada una de las partes del hardware (monitor, mouse, teclado, impresora, CPU, CD-ROM, disco duro, etc.).

Es por esto que el software puede ser considerado como un tipo de interfaz entre el hardware, los datos que vienen incorporados en el mismo y el mismo software u otros, también puede contener información, para que otros software, puedan ser ejecutados. Nunca se nos tiene que olvidar, que el software, es información pura, la cual contiene instrucciones, que deben ser ejecutadas tanto por el hardware u otro software. Por lo mismo, es que se dice, que todo software contiene entradas y salidas, ya que la información recircula de manera constante, entre el mismo software y el hardware u otro software.

La mayoría del software lleva consigo, a los servidores, los sistemas operativos, los windowing, drivers de dispositivo entre otras ejecuciones a realizar.

Ahora, se puede hablar de tres tipos de software importantes, el de uso, el de programación y el de sistema. La clase de software que se mencionó en los

párrafos anteriores se refiere más bien al software interno o de sistema. En cuanto al software de uso, esto es orientado al usuario, la verdad es que todos lo utilizamos sin saber lo que en realidad esto incluye a todos los programas populares, como procesadores de texto, bases de datos, juegos y otros.

Ahora, la mayoría de los software, deben ser adquiridos de manera independiente al hardware, por ejemplo, la mayoría del software de uso, deben ser adquiridos en tiendas especializadas de computación. Claro que los hardware, vienen con ciertos software para poder hacerlos funcionar, cómo es el caso de los famosos drivers para impresoras y otros periféricos. Afortunadamente, las nuevas versiones de los sistemas operativos como Windows ya traen instalados la mayoría de los drivers y programas necesarios.

Nosotros debemos de señalar, que el software es uno de los negocios más rentables en la actualidad. Ya que estos de manera constante, van evolucionando y sacando mejoras, para ser utilizadas en los hardware.

Para tener una idea, el software que mayormente debe ser comercializado y conocido, es Windows de Microsoft, este es un típico software de uso u operación.

Hoy en día, no existe una computadora que no utilice Windows, el cual es el más utilizado en la actualidad.

### **4.3 ARDUINO**

Es una plataforma de electrónica abierta para la creación de prototipos basada en software y hardware flexible y fácil de usar, se creó para artistas, diseñadores, aficionados y cualquiera interesado en crear entornos u objetos interactivos.

Arduino puede tomar información del entorno a través de sus pines de entrada de toda una gama de sensores y puede afectar aquello que le rodea controlando luces, motores y otros actuadores. El microcontrolador en la placa Arduino se

programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador, si bien tienen la posibilidad de hacerlo y comunicar con diferentes tipos de software.

Arduino tiene diferentes herramientas y códigos que permiten enlazar proyectos en los cuales se intervienen la parte del software con el hardware, estas herramientas son de un fácil uso para todos los usuarios.

Algunas herramientas importantes que debemos de tener en cuenta para la elaboración de proyectos deben de ser: Entradas /salidas digitales, Entradas /salidas analógicas, Estructuras de control.

Figura 1. Arduino



#### 4.4 LEDS

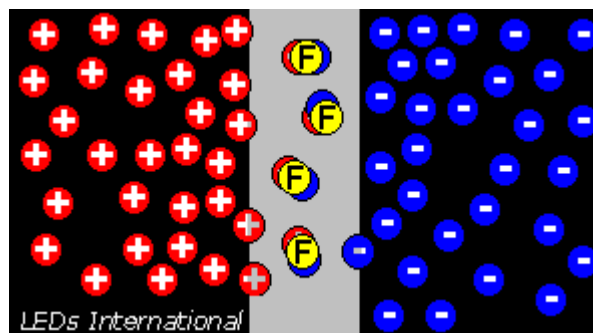
Los LEDs son componentes eléctricos semiconductores (diodos) que son capaces de emitir luz al ser atravesados por una corriente pequeña. Las siglas “LED” provienen del acrónimo en inglés “Light Emitting Diode” o lo que traducido al español sería “Diodo Emisor de Luz”.



Estos están conformados básicamente por un chip de material semiconductor dopado con impurezas, las cuales crean conjunciones del tipo P-N. Los LEDs, a diferencia de los emisores de luz tradicionales, poseen polaridad (siendo el ánodo el terminal positivo y el cátodo el terminal negativo) por lo que funcionan únicamente al ser polarizados en directo.

La electroluminiscencia se da cuando, estimulados por un diferencial de voltaje en directo sobre sus terminales, las cargas eléctricas negativas (electrones) y las cargas eléctricas positivas (huecos) son atraídas a la zona de conjunción donde se combinan entre sí, dando como resultado la liberación de energía en forma de fotones como se ilustra en la siguiente figura:

Figura 2. Funcionamiento físico de los LEDs



Esto da como resultado una generación de luz mucho más eficiente ya que la conversión energética se da con mucho menos pérdida en forma de calor como ocurre con bombillas regulares con resistencias.

Los LEDs son componentes que, dependiendo de la combinación de los elementos químicos presentes en los materiales que los componen, (ejemplo: AlGaInP, GaAs, GaInN, GaP, etc.) pueden producir un amplio rango de longitudes de onda dentro del espectro cromático, dando como resultado diferentes colores,

desde el infrarrojo, pasando por todo el abanico del espectro visible (rojos, amarillos, verdes, azules), hasta ultravioleta, por lo que son muy versátiles en cuanto a su uso en aplicaciones que requieren fuentes de iluminación con longitudes de onda que no se habían podido obtener previamente con fuentes de luz tradicionales.

Figura 3. Leds en su forma física



Son muchas las ventajas que poseen los LEDs ante los dispositivos tradicionales de iluminación como bombillos incandescentes, alógenos, tubos de neón, etc.

Reducen significativamente el consumo energético en comparación a las luminarias tradicionales tales como los bombillos incandescentes, alógenos, entre otros.

Tiempo estimado de vida muy elevado, por lo que se reducen costos de mantenimiento.

Trabajan a muy baja corriente y tensión lo que los hace más seguros y confiables en su implementación.

Virtualmente no generan calor (cuando son implementados a baja potencia).

Por ser de estado sólido pueden ser adaptados a aplicaciones con ciertos grados de vibraciones o impactos.

Son excelentes para ser implementados en sistemas micro-controlados o con niveles de tensión TTL por trabajar a bajo voltaje.

Tiempo de respuesta ON/OFF – OFF/ON virtualmente instantáneo

Puede ajustarse su intensidad en el brillo por medio de modulación en frecuencia.

Son ideales para el diseño de dispositivos de iluminación multicolor.

Permiten la elaboración de dispositivos de iluminación mucho más prácticos y de fácil instalación.

En términos generales, pueden considerarse de forma aproximada los siguientes valores de diferencia potencial:

Rojo = 1,8 a 2,2 voltios.

Anaranjado = 2,1 a 2,4 voltios

Amarillo = 2,1 a 2,4 voltios.

Verde = 2 a 3,5 voltios.

Azul = 3,5 a 3,8 voltios.

Blanco = 3,6 voltios.

## 4.5 LOS CIRCUITOS INTEGRADOS

El ULN2803A es un dispositivo consta de ocho pares de Darlington NPN que cuentan con salidas de alto voltaje con abrazadera de cátodo común, diodos para la conmutación de cargas inductivas. La calificación colector de corriente de cada par Darlington es de 500 mA, pueden estar conectados en paralelo para una mayor capacidad de corriente.

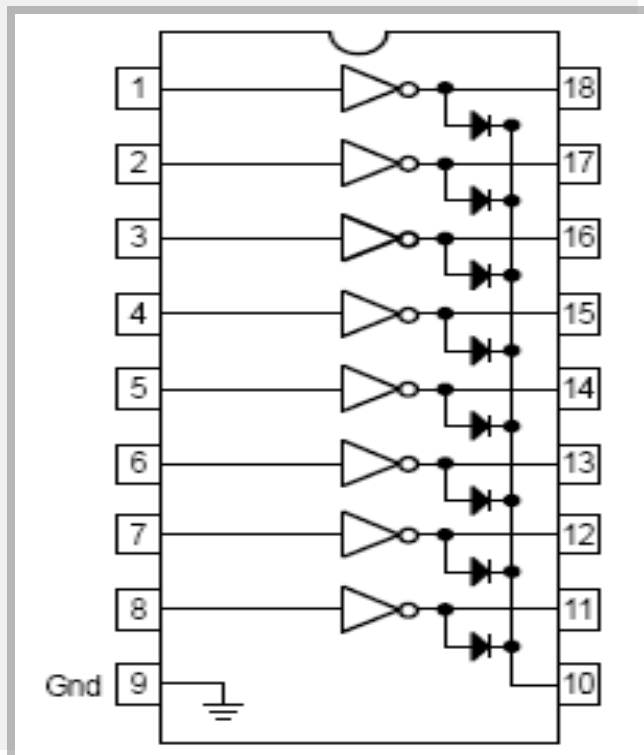
Las aplicaciones incluyen conductores, controladores de relé de martillo, conductores, controladores de pantalla de la lámpara (LED y descarga de gas), línea de drivers y tampones lógica. El ULN2803A tiene una resistencia en serie de 2,7 kW base para cada par Darlington para el funcionamiento directamente con TTL o 5-V dispositivos CMOS.

En un Control hemos utilizado este integrado en algunos proyectos, como en el Módulo Relés x 8 que forma parte del PIC TRAINER.

Figura 4. Circuito integrado, aspecto físico.



Figura 5. Conexión del circuito integrado ULN 2803A



## **5. METODOLOGÍA**

### **5.1 TIPO DE PROYECTO**

Es un diseño de un tablero public para mejorar la información dentro de la institución.

### **5.2 TIPO DE INVESTIGACIÓN**

Aplicada: es aquella que va encaminada a la producción de bienes y servicios.

### **5.3 MÉTODO**

Deductivo porque se parte de la necesidad de mejorar la información de la institución con tableros pasa mensajes que se podrán estar modificando para cualquier cambio de información.

### **5.4 TÉCNICAS DE RECOLECCIÓN DE LA INFORMACIÓN**

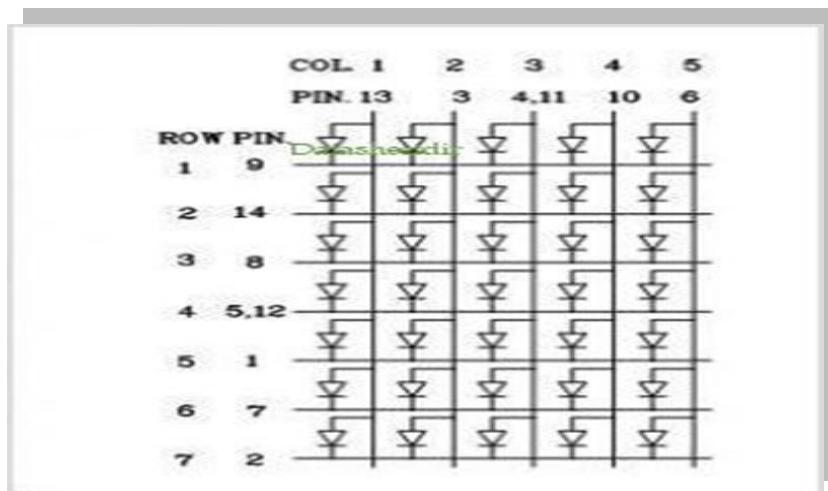
**5.4.1 Fuentes primarias:** Entrevista a profesores y alumnos sobre la falta de información en carteles de la institución y se concluyó que este es un gran problema por la pérdida de tiempo y de gastos en papelería sobre lo que ocurre diariamente en los corredores y bloques.

**5.4.2 Fuentes secundarias:** Libros técnicos, revistas técnicas, catálogos e internet.

## 6. RESULTADOS DEL PROYECTO

El proyecto que se va a elaborar es una pantalla de leds programable en el cual se podrá mostrar textos programados previamente, para lograr esto se hará a través de una serie de filas y columnas los cuales darán la luz a las letras y posteriormente a las palabras, y luego con la unión de varias palabras formaran una frase la cual da la ilusión de desplazarse lentamente. En este proyecto se utilizara tanto la parte de software como de hardware ya que ambas serán necesarias para la elaboración de dicho proyecto. El proyecto estará funcionando correctamente una vez se fusione, tanto la parte del software como la del hardware, para poder obtener el resultado necesario. En este caso estamos utilizando una matriz de leds 5x7 monocromo, 5 resistencias de 330 ohmios (para limitar la corriente en las 5 columnas, cátodo en cada fila) y un Arduino Mega. Físicamente tenemos en este componente 14 pines (7 en cada lado), pero debemos saber leer qué pines representan las columnas y filas. Una Web confiable para esto es Datasheet, a continuación la distribución correspondiente para la matriz de LEDS:

Figura 6. Matriz de LEDS



Se observa en el gráfico que el pin 13 corresponde a la conexión de la columna 1, el pin 3 a la columna 2, el pin 4 y 11 representan a la columna 3 y así respectivamente; pero cómo podemos determinar cuál es el pin 13, 3, 4 u 11 físicamente, para ello, testeamos cada pin con un multímetro, este caso el primer pin de la parte superior derecha representa el pin 1 y el último pin de la derecha representa el pin 14.

Se procede a explicar y mostrar la programación para el funcionamiento del tablero pasa mensajes publik. En primer lugar el programa completo se describe cada paso de programación comienza con poner todos los pines como salidas digitales:

```
int pin[80] = {1,1,2,3,4,5,6,7,8,9,10,11,12,13,24,25,26,27,28,29,30,31,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47,48};
```

Describimos cuales son los pines que manejan las salidas de las columnas:

```
int cols[48] = {pin[9],pin[10],pin[11],pin[12],pin[13],pin[14],pin[15],pin[16],pin[17],pin[18],pin[19],pin[20],pin[21],pin[22],pin[23],pin[24],pin[25],pin[26],pin[27],pin[28],pin[29],pin[30],pin[31],pin[32],pin[33],pin[34],pin[35],pin[36],pin[37],pin[38]};
```

Describimos cuales son los pines que manejan las filas las cuales van unidas en todas las matrices por eso se ponen 7 salidas:

```
int fils [7] = { pin[2], pin[3], pin[4], pin[5], pin[6], pin[7], pin[8]};
```

Definimos los números decimales que hacen falta para dibujar cada carácter:

```
#define PP {0}
#define SP {0, 0, 0, 0, 0} //espacio
#define EX {0, 125, 0, 0, 0} //exclamación
#define A {31, 36, 68, 36, 31}
#define B {127, 73, 73, 73, 54}
#define C {62, 65, 65, 65, 34}
```



```
#define D {127, 65, 65, 34, 28}
#define E {127, 73, 73, 65, 65}
#define F {127, 72, 72, 72, 64}
#define G {62, 65, 65, 69, 38}
#define H {127, 8, 8, 8, 127}
#define I {0, 65, 127, 65, 0}
#define J {2, 1, 1, 1, 126}
#define K {127, 8, 20, 34, 65}
#define L {127, 1, 1, 1, 1}
#define M {127, 32, 16, 32, 127}
#define N {127, 32, 16, 8, 127}
#define EN {31, 72, 68, 66, 31} //eñe
#define O {62, 65, 65, 65, 62}
#define P {127, 72, 72, 72, 48}
#define Q {62, 65, 69, 66, 61}
#define R {127, 72, 76, 74, 49}
#define S {50, 73, 73, 73, 38}
#define T {64, 64, 127, 64, 64}
#define U {126, 1, 1, 1, 126}
#define V {124, 2, 1, 2, 124}
#define W {126, 1, 6, 1, 126}
#define X {99, 20, 8, 20, 99}
#define Y {96, 16, 15, 16, 96}
#define Z {67, 69, 73, 81, 97}
#define CR {24, 60, 30, 60, 24} //corazón
#define CF {4, 50, 2, 50, 4} //cara feliz
#define DP {0, 20, 0, 0, 0}
#define GN {0, 8, 8, 8, 0} //guion
#define NUM {20, 127, 20, 127, 20} //sigla numero
```

```

#define PUN {0, 1, 0, 0, 0} //punto
#define INA {6, 9, 89, 2, 0} //signo de interrogación abierto
#define INB {0, 32, 77, 72, 48} //signo de interrogación cerrado
#define PES {18, 42, 127, 42, 36} //signo peso
#define UNO {0, 33, 127, 1, 0}
#define DOS {1, 51, 69, 73, 49}
#define TRE {34, 65, 73, 73, 54}
#define CUA {12, 20, 36, 127, 4}
#define CIN {121, 73, 73, 73, 70}
#define SEI {62, 73, 73, 73, 6}
#define SIE {64, 64, 79, 80, 96}
#define OCH {54, 73, 73, 73, 54}
#define NUE {48, 73, 73, 73, 62}
#define CER {62, 69, 73, 81, 62}
#define FLE {15, 7, 15, 29, 24} //flecha

```

Se define el cambio de velocidad dando como el 1 la velocidad mayor y el 7 la velocidad menor:

```

int dispSpeed = 1;
byte bit [7] = {64, 32, 16, 8, 4, 2,1};

```

```

byte colVals [30] = {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1};

```

Para este caso se coloca lo que se desea escribir en letras mayúsculas dejando una coma intermedia por cada letra número o carácter diferente, ya colocando la frase deseada se procede a contar el número de caracteres y se cambia por el 8:

ejemplo:

```
const int charNum = 8;
byte string[charNum][5] = {PP,D,A,I,R,O,N,PP};
```

Programamos la placa:

```
void setup () {
  Serial.begin (9600);
```

Ponemos a nivel alto todas las líneas de los cátodos de la matriz:

```
for (int i = 2; i < 48; i++) {
  pin Mode (pin[i], OUTPUT);}
```

Se coloca un pequeño contador par a las columnas de las matrices incrementen de a uno:

```
for (int col = 0; col < 30; col++){
  digital Write (cols[col], LOW);}
for (int fil = 1; fil < 7; fil++){
  digital Write (fils[fil], LOW);
}
  Serial.println (availableMemory ());
}
```

```
void loop () {
  for (int ltr = 0; ltr < charNum; ltr++) {
    for (int y = 0; y < 6; y++) {
      shifLeft ();
      if (y < 5) {
        colVals[30] = string[ltr][y];}
      else {colVals[30] = 0;}
      for (int x = 0; x < dispSpeed; x++) {
        display ();
      }
    }
  }
}
void shifLeft (){
```

```

for (int x = 0; x < 30; x++){
colVals[x] = colVals[x + 1];
}
}
void display () {
for (int col = 0; col < 30; col++){
for (int fil = 0; fil < 7; fil++) {
if (colVals[col] & bit[fil]){
digital Write(fil[s[fil]], HIGH);}
}
digital Write (cols[col], HIGH);
delay (1);

```

Se coloca un pequeño contador para que las filas de las matrices incrementen de a uno

```

for (int fil = 0; fil < 7; fil++){
digital Write (fil[s[fil]], LOW);
}
digital Write(cols[col], LOW);
}
}
int availableMemory() {
int size = 1024;
byte *buf;
while ((buf = (byte *) malloc(--size)) == NULL);
free(buf);
return size;
}

```

Lo que hemos hecho es definir cada letra del abecedario y los números por una serie de números decimales. Estos números decimales son los que mandaremos a nuestros registros de desplazamiento para decirle que columnas de diodos LED debe iluminar.

Debemos imaginar que el primer diodo LED vale 1 ( $2^0$ ), el segundo 2 ( $2^1$ ), el tercero 4 ( $2^2$ ) y así sucesivamente, de manera que los que necesitemos tener encendidos, deberemos sumarlos entre sí.

Ahora, para dibujar una letra o símbolo, tan sólo hace falta tiempo y paciencia! Por ejemplo, nuestras letras son de 7 columnas x 5 filas, de manera que la letra O por ejemplo sería:

```
#define O {62, 65, 65, 65, 62}
```

Que al pasarlo a binario nos quedaría así:

```
62 =>0 1 1 1 1 1 0  
65 =>1 0 0 0 0 0 1  
65 =>1 0 0 0 0 0 1  
65 =>1 0 0 0 0 0 1  
62 =>0 1 1 1 1 1 0
```

Y donde tenemos un 1, tenemos un diodo.

Después de tener toda la parte del software explicada y programada se procede conectar toda la parte del hardware de la siguiente forma:

Al tener claramente cuáles son los pines de salida y que maneja cada uno se procede a conectar los pines del Arduino a los circuitos integrados ULN2803A los cuales en la primera matriz lo manejan dos circuitos integrados uno para las columnas y otro para las filas. Las filas se conectan en paralelo para todas las matrices para solo utilizar 7 salidas , las columnas se conectan de tal forma que la primera matriz la manejan los últimos pines del Arduino y así sucesivamente se van conectando en orden descendente.

## **7. CONCLUSIONES**

Tener parte activa en la elaboración del proyecto, ha permitido poner en práctica conocimientos adquiridos a lo largo del plan de estudio de la Tecnología en Electrónica en la institución universitaria pascual bravo; para así brindar un aporte en las nuevas tecnologías.

Se logra incursionar en un campo de la publicidad luminosa como lo son los tableros públicos y en la programación de estos para crear publicidad luminosa e información.

Utilizar tecnologías nuevas como lo son los microcontroladores Arduinos que son fáciles de programar y con infinidad de aplicaciones en la electrónica.

Una herramienta de gran valor para la realización de este prototipo, ha sido el aporte en conocimiento de muchas personas de diferentes campos como: la electrónica, programación, entre otras; para poder explorar opciones, que han permitido dar solución a dificultades presentadas en el desarrollo de este.

Es de resaltar la importancia que ha tenido el uso de la Internet como fuente de consulta en los temas tratados, para la consecución de información que ha permitido enriquecer la documentación del proyecto.

## **8. RECOMENDACIONES**

Se debe tener un debido cuidado con este proyecto ya que cuando se va a realizar el cambio de las frases que se desean poner para que aparezcan en el tablero se puedan borrar instrucciones del microcontrolador.

Consultar y seguir las instrucciones y recomendaciones para el correcto funcionamiento del prototipo.

El cebado de la bomba peristáltica, que es la recarga del líquido a transportar, se debe hacer por aparte mediante el uso de los pulsadores indicados para tal propósito y previamente colocar recipiente en la boquilla de dosificación para evitar derramar el producto.

Antes de poner el sistema en operación, se debe tener la precaución de compilar en el software y cargarlo debidamente para el buen funcionamiento del prototipo.

Se debe ubicar la tablero en un lugar seguro, protegido de la humedad para evitar el deterioro de sus partes.

Posicionar el tablero pasa mensajes publik en un lugar visible y con la posibilidad de tener un toma eléctrico de 110v para poder proveer el tablero de energía para su funcionamiento.

Este proyecto es para uso de información y de publicidad dentro de la institución dejando aúna persona encargada de la manipulación de este con una buen capacitación del funcionamiento.

## CIBERGRAFÍA

Neoteo. Matriz de leds8x8 (En línea)<http://www.neoteo.com/matriz-de-l>

Neoteo. Matriz de leds8x8 (En línea)<http://www.neoteo.com/matriz-de-led-8x8>  
[citado en Abril de 2012]

Indicat Carteles electrónicos Manual de operación. [En Línea]  
<http://www.indicat.om.ar/manualL130.html> [Citado en Abril de 2012]

UControl Matriz de LEDS de 7x32 (en Línea)  
[http://www.ucontrol.com.ar/wiki/index.php/MAtriz\\_de\\_leds\\_de\\_7x32](http://www.ucontrol.com.ar/wiki/index.php/MAtriz_de_leds_de_7x32)

[Citado en Abril de 2012]

Palazzesi, Ariel. Matrices de leds Todo lo que tienes que saber para la construcción de tu propio cartel de matriz de LED,s Mayo de 2008

[http://www.ucontrol.com.ar/revista/0003/ucontrol\\_revista\\_0003.pdf](http://www.ucontrol.com.ar/revista/0003/ucontrol_revista_0003.pdf)

[Citado en Mayo de 2012]

DBup. Desarrollos electrónicos basados en leds (En Linea)

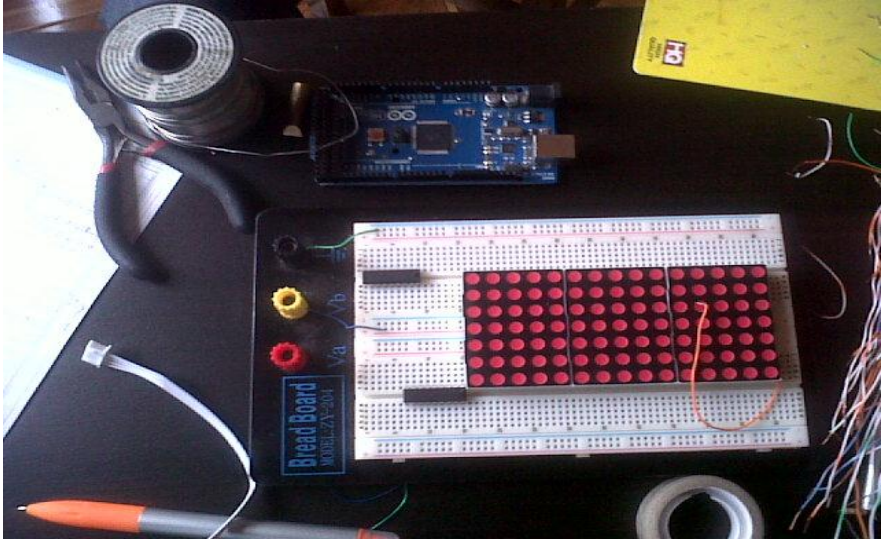
<http://www.bdup.com.ar/> [Citado en Mayo de 2012]

Arduino. Plataforma para la creación de hardware sencillo (Citado en línea)

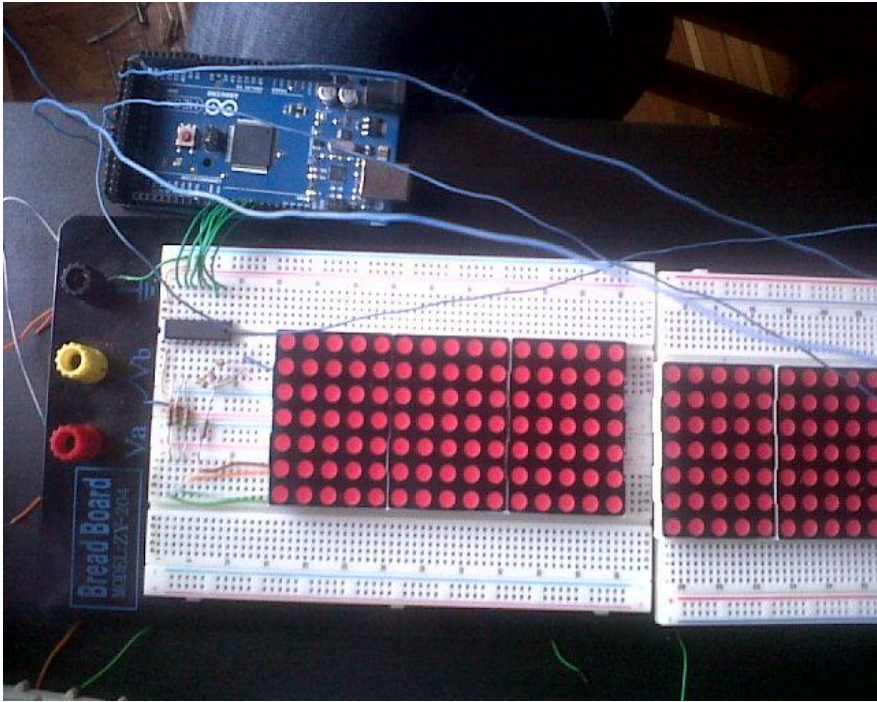
<http://arduino.cc/es/> [Citado en Mayo de 2012]



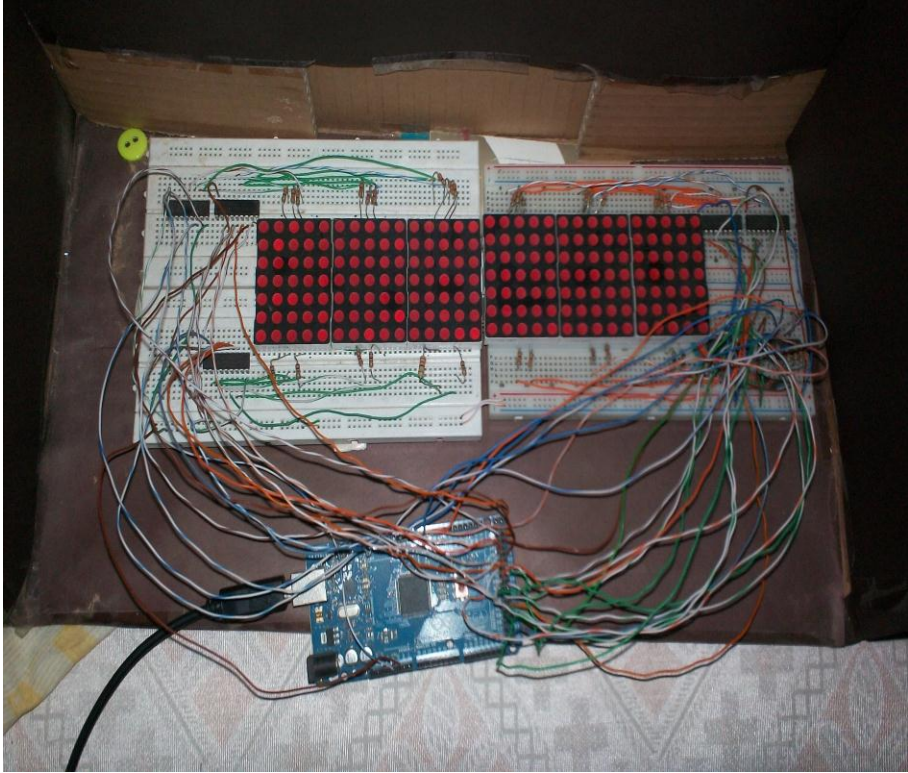
**Anexo A: montaje 1**



**Anexo B: montaje 2**



**Anexo C: montaje 3**

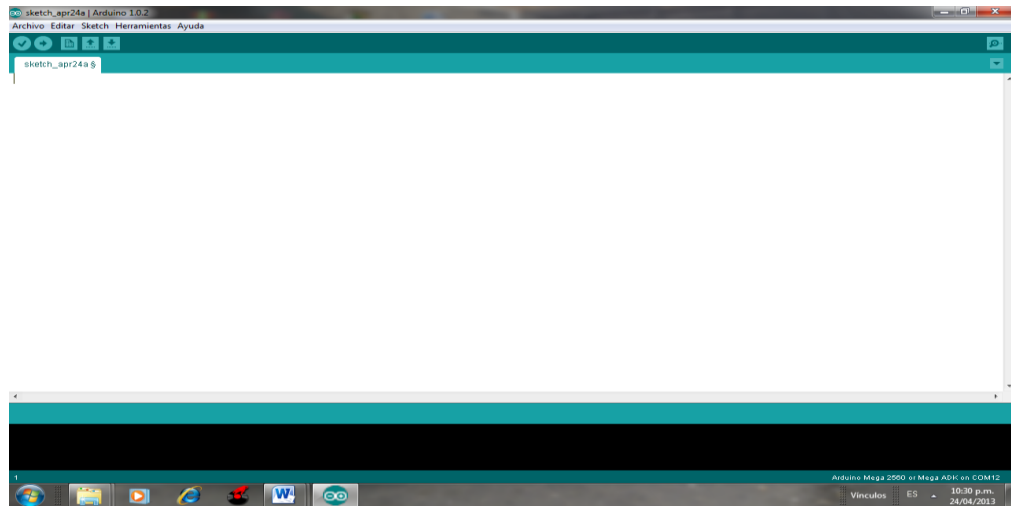


## Anexo D: Manual técnico

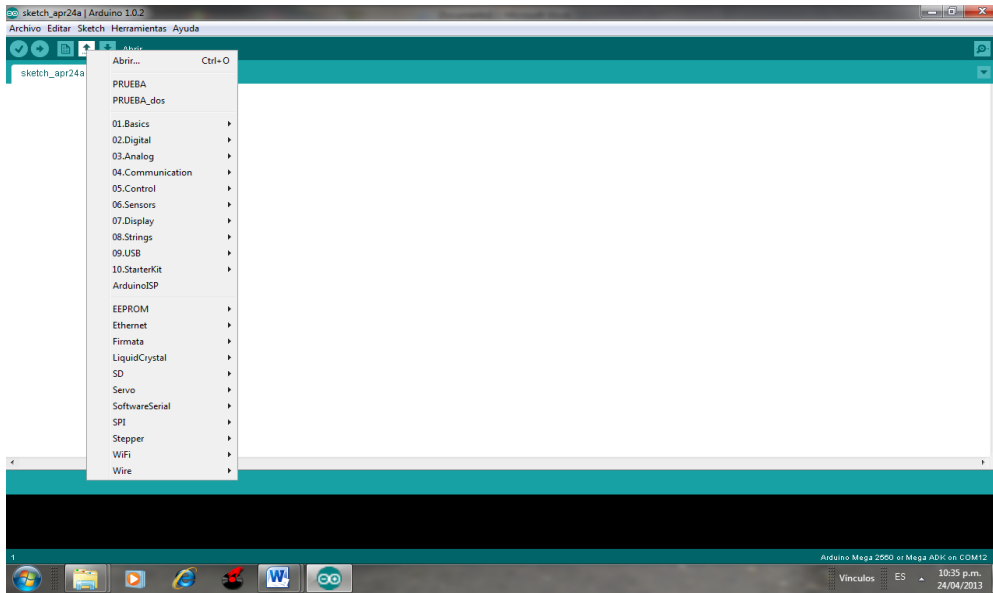
### Instalar y programar su primer mensaje

Descargue la aplicación desde <http://www.arduino.com> ejecute el programa siga los pasos sugeridos por el programa de instalación. Normalmente presionando ENTER, en todas las ventanas de instalación es suficiente para realizarla correctamente.

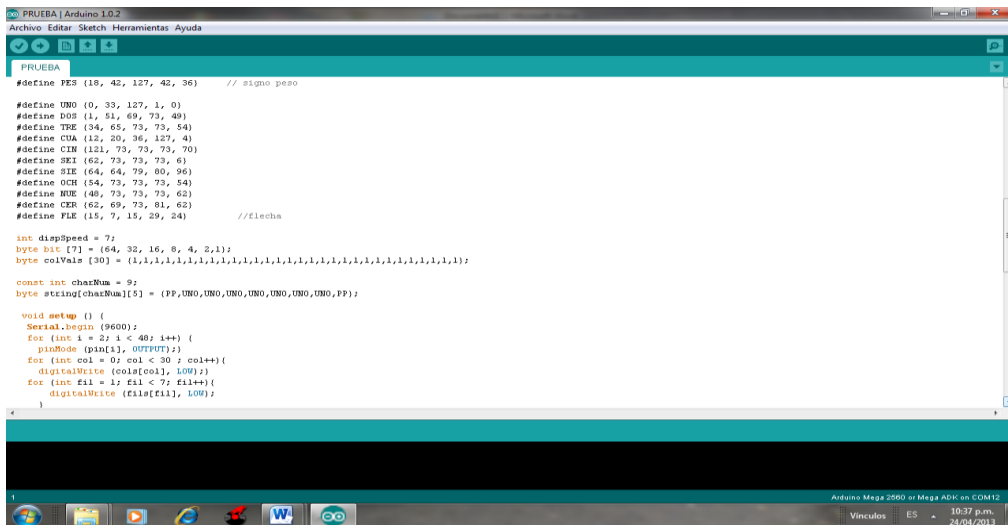
Luego de instalado el programa le damos inicio y abrimos el programa de arduino.



Después de tener abierto el programa llevamos el cursor a la parte de abrir



Seguimos el cursor ala parte de prueba y abrimos el programa.



En la parte donde esta señalado con el circulo colocamos el mensaje que queremos que aparezca en el tablero con las letras mayúsculas y separada cada letra con comas dejando siempre "PP" siempre al principio y al final como lo vemos en este ejemplo (PP,P,R,O,Y,E,C,T,O,D,E,G,R,A,D,O,PP) y donde se

encuentra el circulo pequeño colocamos el número de caracteres que estamos utilizando, **const int charNum = 17;**

```
PRUEBA
#define PES (18, 42, 127, 42, 36) // signo peso

#define UNO (0, 33, 127, 1, 0)
#define DOS (1, 51, 69, 73, 49)
#define TRE (34, 65, 73, 73, 54)
#define CUA (12, 20, 36, 127, 4)
#define CIN (121, 73, 73, 73, 70)
#define SEI (62, 73, 73, 73, 6)
#define SIE (64, 64, 79, 80, 96)
#define OCH (54, 73, 73, 73, 54)
#define NUE (48, 73, 73, 73, 62)
#define CER (62, 69, 73, 81, 62)
#define FLE (15, 7, 15, 29, 24) //flecha

int dispSpeed = 7;
byte bit [7] = {64, 32, 16, 8, 4, 2,1};
byte colVals [30] = {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1};

const int charNum = 9;
byte string[charNum][8] = {"PP,UNO,UNO,UNO,UNO,UNO,UNO,PP}";

void setup () {
  Serial.begin (9600);
  for (int i = 2; i < 48; i++) {
    pinMode (pin[i], OUTPUT);}
  for (int col = 0; col < 30 ; col++){
    digitalWrite (cols[col], LOW);}
```

Después de escribir el mensaje que queremos que aparezca en la pantalla de led nos dirigimos en la parte superior de la izquierda y le damos verificar, esperamos un momento y el programa nos dirá que está listo para cargar.

```
PRUEBA | Arduino 1.0.2
Archivo Editar Sketch Herramientas Ayuda

PRUEBA
#define INA (6, 9, 89, 2, 0) //sig interrogacion a
#define INB (0, 32, 77, 72, 48) //sig interrogacion b
#define PES (18, 42, 127, 42, 36) // signo peso

#define UNO (0, 33, 127, 1, 0)
#define DOS (1, 51, 69, 73, 49)
#define TRES (34, 65, 73, 73, 54)
#define CUA (12, 20, 36, 127, 4)
#define CIN (121, 73, 73, 73, 70)
#define SEI (62, 73, 73, 73, 6)
#define SIE (64, 64, 79, 60, 96)
#define OCH (54, 73, 73, 73, 54)
#define NUE (48, 73, 73, 73, 62)
#define CER (62, 69, 73, 81, 62)
#define FLE (15, 7, 15, 29, 24) //flecha

int dispSpeed = 7;
byte bit [7] = {64, 32, 16, 8, 4, 2, 1};
byte colVals [30] = {1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1,1};

const int charNum = 9;
byte string[charNum][5] = {PP,UNO,UNO,UNO,UNO,UNO,UNO,UNO,PP};

void setup () {
  Serial.begin (9600);
  for (int i = 2; i < 48; i++) {
    pinMode (pin[i], OUTPUT);
  }
  for (int col = 0; col < 30 ; col++){
    digitalWrite (cols[col], LOW);
  }
  for (int fil = 1; fil < 7; fil++){
```

Compilación terminada

Tamaño binario del Sketch: 6.252 bytes (de un máximo de 258.048 bytes)

1 Arduino Mega 2560 or Mega ADK on COM12 ES 09:19 p.m. 25/04/2013

Después de verificar que el programa este bien y no resulte con ningún error, conectamos el tablero por medio de un cable USB al computador, con el cursor nos dirigimos a la flecha don dice cargar le damos clic y ya queda cargado el mensaje anteriormente escrito en el tablero.

Ya que tengamos el mensaje cargado en el tablero procedemos a conectarlo a una fuente de voltaje de 110v con el adaptador con el que viene previamente el tablero, y ya con esto podremos mirar en funcionamiento del tablero pasa mensajes

