

UNIVERSIDAD NACIONAL DE LA MATANZA

Departamento de Ingeniería e Investigaciones Tecnológicas

**Ingeniería en Electrónica**

**Técnicas digitales III**

Actualización de troqueladora manual de mandíbulas

**Alumno:**  Mateo Ignacio Serrano

**Profesores:** Lic. Carlos Maidana

Ing. Guillermo Buranits

Ing. Mauro Cipollone

2018

**ÍNDICE**

[1 Descripción general 1](#_Toc26440533)

[2 Introducción teórica 2](#_Toc26440534)

[3 Descripción técnica del proyecto 3](#_Toc26440535)

[3.1 Mecánica 3](#_Toc26440536)

[3.2 Hardware 5](#_Toc26440537)

[3.2.1 Diagrama en bloques 5](#_Toc26440538)

[3.2.2 Esquemático 5](#_Toc26440539)

[3.2.3 Circuito impreso 8](#_Toc26440540)

[3.2.4 Fotos 9](#_Toc26440541)

[3.3 Software 9](#_Toc26440542)

[3.3.1 Enumeración de rutinas 9](#_Toc26440543)

[3.3.2 Descripción del funcionamiento de cada rutina 11](#_Toc26440544)

[3.3.2.1 Rutina “main”: 11](#_Toc26440545)

[3.3.2.2 Máquinas de estados y rutina sensor 13](#_Toc26440546)

[3.3.2.3 Subrutina “HAL\_Init”: 16](#_Toc26440547)

[3.3.2.4 Subrutina “SystemClock\_Config”: 16](#_Toc26440548)

[3.3.2.5 Subrutinas “MX\_GPIO\_Init, SPI1\_Init, TIM4\_Init”: 16](#_Toc26440549)

[3.3.2.6 Subrutina “LCD\_Ini”: 17](#_Toc26440550)

[3.3.2.7 Subrutina “MFRC522\_Init”: 18](#_Toc26440551)

[3.3.2.8 Subrutina “MY\_FLASH\_SetSectorAddrs”: 18](#_Toc26440552)

[3.3.2.9 Subrutina “flash\_cargar\_operarios”: 19](#_Toc26440553)

[3.3.2.10 Subrutina “display\_escribir”: 21](#_Toc26440554)

[3.3.2.11 Subrutina “encuesta\_teclado”: 22](#_Toc26440555)

[3.3.2.12 Subrutina “conv\_hex”: 22](#_Toc26440556)

[3.3.2.13 Subrutina “MFRC522\_Check”: 23](#_Toc26440557)

[3.3.2.14 Subrutina “MFRC522\_Compare”: 23](#_Toc26440558)

[3.3.2.15 Subrutina “flash\_guardar\_operarios”: 23](#_Toc26440559)

[3.3.2.16 Subrutina de interrupción “HAL\_GPIO\_EXTI\_Callback”: 23](#_Toc26440560)

[3.3.2.17 Subrutina “delayus\_block”: 24](#_Toc26440561)

[3.3.2.18 Subrutina “introducir\_texto”: 24](#_Toc26440562)

[3.3.2.19 Subrutina “display\_unidades”: 25](#_Toc26440563)

[4 Modo de operación 25](#_Toc26440564)

[4.1 Inicio del equipo 25](#_Toc26440565)

[4.2 Teclado 25](#_Toc26440566)

[4.3 Menús 26](#_Toc26440567)

[4.4 Operación 27](#_Toc26440568)

[5 Ensayos 27](#_Toc26440569)

[6 Conclusiones 28](#_Toc26440570)

[7 Proyecto finalizado 29](#_Toc26440571)

[8 Referencias 30](#_Toc26440572)

# Descripción general

¿Qué tienen en común una caja de pizza, un mazo de cartas, un rompecabezas y una caja de medicamentos?... Además de algún dato curioso que se nos pueda ocurrir, lo que comparten es el proceso productivo con el que se obtienen tales productos.

El troquelado es el método por excelencia para la fabricación de diversos productos de cartón, cartulina e incluso plástico en láminas muy finas. El mismo permite que a partir de una hoja o lámina plana, se realicen cortes y/o marcaciones para pliegues que determinarán el tamaño y la forma del producto o le darán rigidez estructural si es necesario.

Existen máquinas troqueladoras automáticas y manuales, siendo estas últimas las más comunes y las más antiguas del mercado local. Teniendo en cuenta esto, es de esperar que esta maquinaria sea algo rústica y carezca de sistemas de asistencia al operario.

El objetivo del proyecto en cuestión es brindar una herramienta accesoria a una troqueladora manual útil tanto para el operario, como para el dueño de la fábrica o el encargado de producción.

El sistema permitirá a cada operario iniciar su propia sesión de trabajo en la máquina mediante el uso de tarjetas personales, esto permitirá al superior responsable llevar un registro de la producción de cada uno. Además de asegurar que la máquina no podrá ser utilizada por usuarios que no cuenten con su identificación, es decir, que no estén autorizados a operarla.

Por otro lado, se podrá visualizar en el sistema la cantidad de unidades que fueron troqueladas en cada trabajo. Esto provee una mejora sustancial para el usuario, ya que no tiene que estar pendiente del conteo de hojas que pasa por la máquina mientras trabaja, y de esa manera, prestar más atención a su seguridad personal.

Además de observar el número de unidades procesadas, se podrá verificar la cantidad de ciclos de la máquina que fueron necesarios para producirlas. Esto es un indicador de eficiencia, que será de utilidad al superior responsable para ajustar parámetros como la velocidad de trabajo, los tiempos máximos de utilización de la maquinaria y cualquier otro factor que considere necesario para la optimización del proceso.

El sistema constará de una pantalla y una serie de botones para acceder a la información previamente almacenada y, a su vez, para observar el conteo del trabajo en curso en el momento que está siendo operada la máquina.

# Introducción teórica

A partir de entrevistas con el encargado de una fábrica de troquelados, el análisis mecánico general de la máquina y la observación de operarios realizando sus tareas cotidianas, se llegó a la conclusión de que es necesario discriminar el conteo de los ciclos de apertura y cierre de las mordazas de la máquina de las unidades troqueladas, ya que el movimiento de la máquina no se interrumpe si el operario no introduce el material sobre la mordaza.

Teniendo en cuenta esto, se eligió un sensor inductivo para el conteo de los ciclos de la máquina. Su principio de funcionamiento se basa en generar una señal senoidal a partir de un circuito formado por una bobina y detectar la ampitud de la señal generada. La bobina en cuestión está dispuesta de forma que el campo magnético que genera, se concentre en un área en particular (el área de detección). Al ingresar un material ferroso al área de detección, quedará sometido al campo magnético de la bobina, lo cual inducirá una corriente en el material, provocando una transferencia de energía que causará la disminución de la amplitud de la señal senoidal generada-detectada por el sensor, provocando un cambio en la salida, indicando la presencia de un objeto metálico. Este tipo de sensores son exclusivamente para la detección (o no) de objetos metálicos y la ventaja contra un interruptor mecánico es que, al no tener partes móviles, no tiene desgaste a lo largo del tiempo, por lo tanto, no requiere mantenimiento.

Por otro lado, para detectar la presencia (o no) de material, se utilizará un sensor del tipo infrarrojo. Este tipo de sensores se basan en emitir un haz de luz infrarroja, no visible y, en el mismo dispositivo, implementar un detector para ese tipo de ondas. De esta manera, el emisor genera el haz y el mismo viaja por el aire hasta que se desvanece, pero si el mismo se encuentra con una obstrucción en el camino, la dirección de haz se verá reflejada y alcanzará al detector, de esta manera se interpretará que hay un objeto en la zona de detección. Estos sensores son muy versátiles en cuanto a características del material a detectar, ya que prácticamente cualquier objeto refleja lo suficiente como para interrumpir parte del haz y enviarlo nuevamente hacia el sensor, incluso los transparentes.

En cuanto a la identificación de operarios, debía ser un método sencillo, eficaz y rápido, que no implique una molestia o incomodidad excesiva al trabajador. Por lo tanto, se eligió el método de identificación por radiofrecuencia. Este sistema consta de un lector, que genera un campo electromagnético mediante el cual es capaz de alimentar, leer y transferir información a tarjetas denominadas “tags” sólo con el acercamiento de las mismas al dispositivo. Cada uno de estos tags tiene un código de identificación único y estará asociado a un operario en particular, de esta manera, es posible un inicio de sesión de trabajo personal para cada empleado con sólo aproximar su tag al lector, sin la necesidad de introducir datos por un teclado o alguna interfaz compleja.

Por último, es necesaria una interfaz donde poder visualizar claramente la información almacenada en el sistema, es decir, la cantidad de ciclos de la máquina del trabajo en curso, lo mismo para las unidades producidas, qué operarios tienen permitido operar la máquina, cuál es el número de identificación de la tarjeta personal de cada uno, qué cantidad de trabajos y unidades lleva acumulado cada operario, entre otros datos. Se optó por un visor LCD retroiluminado de 2 líneas de 16 caracteres y un teclado de 16 teclas para poder ingresar datos y desplazarse a través de los menús.

# Descripción técnica del proyecto

## Mecánica

Si bien el proyecto será montado sobre una máquina, el mismo no cuenta con partes mecánicas que interactúen con la troqueladora, sin embargo, dos de los sensores que componen el sistema estarán amurados al chasis de la misma.

Por un lado, el sensor inductivo deberá estar posicionado de tal manera que en la posición extrema de cierre de las mandíbulas la punta de dicho sensor debe quedar a unos 2 o 3 milímetros de la mordaza móvil, por ejemplo, como se esquematiza en la siguiente figura.



Soporte y sensor

Figura 1 - Ubicación del sensor inductivo.

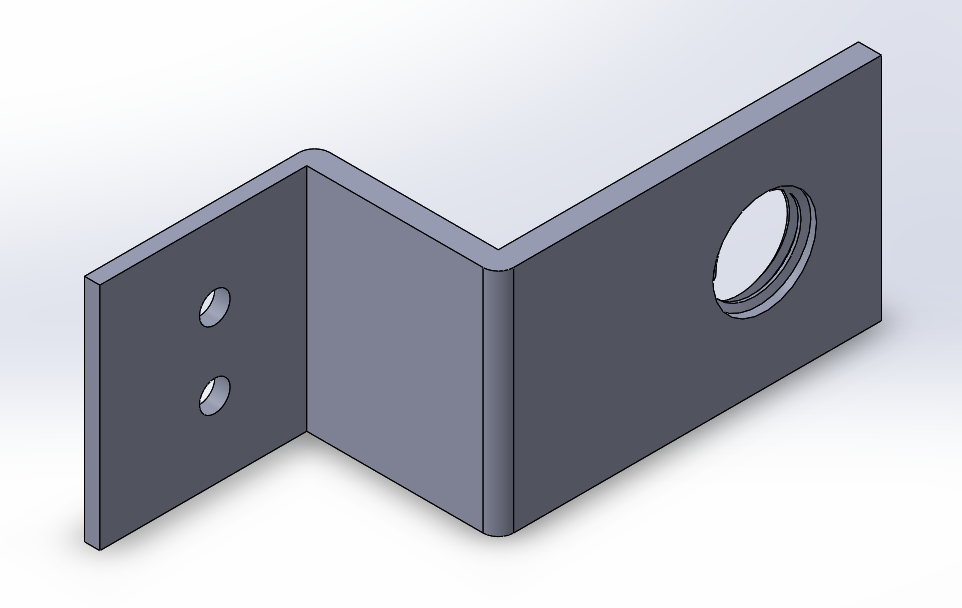


Figura 2 - Soporte del sensor inductivo.

Por otro lado, el sensor infrarrojo que contará las unidades debe estar sujeto al lado izquierdo de la máquina, ya que el material entra por la derecha y se retira por la izquierda, mediante un soporte similar al anterior, pero con el haz apuntando hacia arriba para intersectar la trayectoria de las unidades salientes, como se puede observar en la siguiente figura.



Soporte y sensor

Haz infrarrojo

Figura 3 - Ubicación del sensor infrarrojo.

## Hardware

### Diagrama en bloques

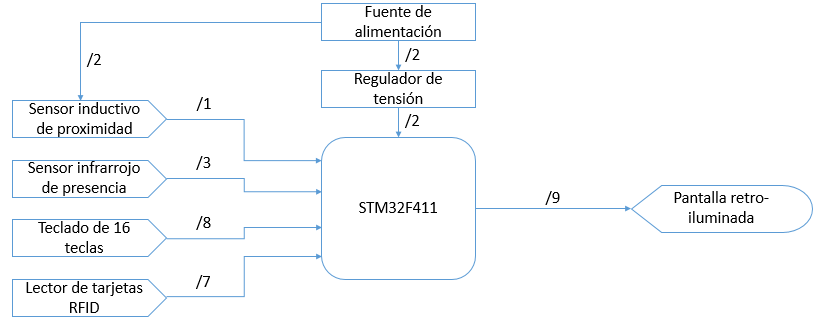


Figura 4 - Diagrama en bloques del hardware

### Esquemático

El sistema completo será alimentado a partir de una fuente de alimentación de 9 V de corriente continua que estará conectada a la tensión de red y su salida será conectada directamente en la placa principal del proyecto. La misma podrá ser una fuente de tensión comercial siempre y cuando sea capaz de entregar al menos 1 A. La justificación principal del uso de esta fuente es el sensor inductivo, el cual requiere ser alimentado con una tensión superior a 6 V, lo cual impide la utilización directa de una fuente con salida de 5 V de corriente continua. En la placa principal habrá dos niveles de tensión de alimentación, el de la fuente en sí y un nivel de 5 V, obtenido a partir de un regulador de tensión integrado (LM7805) para poder alimentar a todos los componentes salvo el sensor susodicho.

El sensor inductivo para el conteo de ciclos de máquina tiene por objetivo generar un pulso cada vez que la máquina se “cierra” para luego volver a abrirse. Este sensor es ideal para tal fin por dos motivos básicos: Por un lado, la estructura de la máquina está prácticamente completamente compuesta por hierro, por ende, dicha estructura es muy permeable ante campos magnéticos externos, es decir, coincide con el principio de funcionamiento del sensor inductivo. La segunda ventaja de este método de detección es que no requiere contacto. Casi toda maquinaria industrial cíclica está expuesta a un número de repeticiones muy elevado, por lo tanto, cualquier dispositivo mecánico susceptible al desgaste, terminaría fallando tarde o temprano. Por otro lado, al observar máquinas más modernas que la troqueladora en la que se basa este proyecto, que poseían un sistema contador integrado de fábrica, se encontró que el tipo de sensor elegido por los fabricantes también era del tipo inductivo. Estos dispositivos poseen una entrada de alimentación y una salida típicamente a colector abierto, lo cual permite trabajar con distintos niveles tensión en la entrada y en la salida. Los parámetros a tener en cuenta a la hora de seleccionar un sensor de este tipo son la distancia de detección, el estado normal (abierto o cerrado) y el tipo de transistor de salida (NPN o PNP). Existe una amplia gama de opciones en el mercado entre los cuales varían, además de los susodichos, las dimensiones del sensor y la tensión de alimentación. El sensor elegido para el proyecto fue el LJ12A3- 4-Z/BX, que posee una distancia de detección de 4 mm, un rango de alimentación de 6 a 36 V, salida normal abierto con transistor NPN y su exterior se encuentra roscado con paso métrico M12, para su montaje es necesario un acople para ubicar el extremo del sensor a menos de 4 mm de la posición límite de la mordaza móvil de la máquina. La salida de este sensor será tratada en el microcontrolador como una entrada de interrupción externa, y la misma deberá ser conectada a través de una resistencia al nivel de tensión de 5 V, debido a su condición de colector abierto, de otra manera, lo que vería la entrada del microcontrolador cuando el sensor esté en estado de no-detección sería una alta impedancia.

Teniendo definido el método de detección del cierre de la máquina, el próximo objetivo fue definir el sensor para el conteo de unidades procesadas. La entrada del proceso son hojas o láminas de un material que puede ser cartulina, cartón, acetato, entre otros de espesor, dimensiones y color variables. El operario siempre toma a mano una de estas hojas, desde una mesa a su lado y la monta sobre la mordaza móvil en un sentido determinado, entonces, la máquina se cierra y luego se vuelve a abrir, momento en el cual, el operario retira la hoja troquelada por el lado contrario al que ingresó, introduce una nueva hoja de material desde la pila inicial y la máquina vuelve a cerrar y abrir. Aprovechando que las hojas troqueladas se retiran siempre por el mismo lado, se decidió montar una barrera infrarroja. La misma poseerá su emisor y transmisor en el mismo punto, y detectará la presencia de un objeto si el mismo reflejó parte del haz de nuevo hacia el sentido de la emisión, lo cual proporciona un montaje más cómodo, ya que se necesita un solo soporte para el sensor y no aporta ninguna limitación en el movimiento de extracción de material de salida de la máquina. Además, los sensores de dos extremos son susceptibles a desalineaciones, que en una máquina en movimiento constante es muy probable que se provoquen con frecuencia. El sensor elegido fue el E18-D50NK, que posee una salida de las mismas características que el inductivo descripto en el párrafo anterior, es decir, con salida normal abierto con transistor NPN a colector abierto, se alimenta con 5 V y posee una distancia de detección ajustable de entre 3 y 50 cm. Será necesario montar este sensor en sentido vertical en algún soporte sujeto a la parte fija de la máquina de modo que, al salir una hoja procesada de la máquina, esta interrumpa el haz. Al igual que para el sensor inductivo, la salida de este sensor será conectada al microcontrolador en una entrada de interrupción externa y con una resistencia de pull-up, por lo explicado anteriormente.

El proyecto también contará con una sección de identificación y registro de operarios. Esto implica la existencia de varios módulos, ya que requiere, al menos, un método de identificación de cada operario, un método de entrada para la configuración del mismo, y un método de salida visual e intuitivo, para poder interactuar con la información registrada por el sistema.

Como primera medida se eligió el método de identificación por RFID, debido a su sencillez, practicidad para el operario y versatilidad. Los “tags” de identificación pueden estar implementados en tarjetas, llaveros o combinaciones de ambos indistintamente para adecuarse a la comodidad del operario. Estos dispositivos son pasivos, es decir que no necesitan una batería para funcionar y tienen la particularidad de poseer un número de identificación único y una especie de memoria donde se pueden almacenar datos no volátiles, es decir, que perduran en el tiempo. En un primer momento, se eligió un módulo de lectura RFID básico (modelo RDM6300) cuya única funcionalidad era leer el tag y enviar por una interfaz UART su Nº de identificación (que representaría a cada operario), lo cual, en principio, sería suficiente ya que el registro de la información de cada uno de los operarios se podría realizar en el controlador principal, sin embargo, a lo largo del desarrollo del proyecto se encontró, habiendo realizado pruebas con distintos microcontroladores y analizando la salida de datos con un osciloscopio sobre dos unidades distintas, que el módulo adquirido no funcionaba. Ante esto, por un tema de confiabilidad, se optó por utilizar el módulo MFRC522, ya que a partir de experiencias anteriores propias y de colegas, no hubo inconvenientes de ningún tipo con su funcionamiento. Este módulo es más complejo que el analizado originalmente, permite leer el Nº de identificación de la tarjeta y también leer y escribir datos en la memoria de la misma, lo cual es una gran oportunidad de mejora para futuras actualizaciones, ya que se descentralizaría el almacenamiento de la información y se podrían agregar muchas más variables de tipo estadístico, por ejemplo. El lector elegido cuenta con un chip con tres interfaces posibles de entrada, I2C, SPI y UART, sin embargo, sólo está cableada a los pines de salida la interfaz SPI, por lo tanto, se utilizará esa para comunicarse con el microcontrolador principal. Esta conexión es sincrónica por lo que, además de las líneas de datos de ida y de vuelta, es necesaria una línea de clock para asegurar el sincronismo entre el dispositivo maestro y el esclavo.

Por otro lado, se necesitaba una manera de visualizar la información, es decir, sin ir más lejos, una pantalla o visor. Inicialmente, se pensó en un conjunto de dígitos de siete segmentos para mostrar la cantidad de golpes y la cantidad de unidades, en conjunto con alguna combinación de LEDs señalizadores para saber cuál de las cantidades estaba siendo indicada a cada momento. Pero al agregar la funcionalidad del registro de operarios, era imperioso que el proyecto posea un sistema de indicación más versátil, en el cual se pueda mostrar muchos más datos y sin ser tan compleja la forma de obtener a partir de ellos, la información correspondiente. Debido a esto, se optó por un visor retro-iluminado con pantalla LCD modelo 1602A-1, el cual posee dos renglones uno encima del otro con 16 caracteres cada uno, se comunica con el microcontrolador principal mediante un protocolo propio del fabricante que consiste en cuatro hilos de datos y una señal de habilitación, además de dos señales de control de lectura/escritura y de instrucción/datos respectivamente. Este dispositivo se alimenta con una tensión de 5 V tanto para la sección de control como para la retro-iluminación y posee una entrada analógica de 0-5 V para regular el contraste de los caracteres respecto del fondo de la pantalla. En el mismo se podrán observar los nombres de los menús, los nombres de los parámetros y el valor de los mismos para cada sección u operario. Todas las entradas de control de la pantalla serán manejadas con salidas GPIO en el microcontrolador principal, tanto los hilos de datos, como la señal de habilitación y las señales de control.

Por último, se necesitaba un método de entrada para poder ingresar al dispositivo los comandos para realizar cada una de las funciones disponibles al usuario, por ejemplo, para recorrer los distintos menús, modificar los datos ingresados de cada operario, cambiar entre la indicación de golpes y la de unidades, etc. la variedad de funciones a realizar para interactuar con el sistema requería una cantidad muy grande de botones, por lo que, en cambio se decidió utilizar un teclado matricial, en este caso de cuatro filas por cuatro columnas, cuyo funcionamiento consiste en alimentar secuencialmente las filas y leer el estado de las columnas y de esta manera, saber si alguna tecla fue pulsada verificando que el valor leído en la columna sea coincidente con el de la fila alimentada. este teclado se comercializa en formato de membrana y en formato placa implementado con pulsadores, optándose por este último debido a que en estos últimos la probabilidad de detección e interpretación errónea de teclas es mucho más baja. En cuanto a la conexión del mismo, los 8 hilos de la matriz serán manejados por pines GPIO del microcontrolador, cuatro de los mismos en configuración de salidas y los restantes en configuración de entradas. Se necesita, además, una resistencia de pull-down para cada una de las líneas de las columnas, para darle un estado bajo “débil” cuando los pulsadores correspondientes a las teclas estén abiertos.

### Circuito impreso

La conexión de todos los elementos contemplados en el apartado anterior decidió llevarse a cabo en un único PCB que contendría los conectores para cada uno de los módulos, sensores y entrada de alimentación.

La premisa principal de este circuito impreso es eliminar todos los falsos contactos, por lo tanto, se utilizaron los conectores más robustos posibles y los cables adecuados para dar prolijidad y confiabilidad al conexionado de los distintos componentes.

En la misma placa principal, se encuentran los componentes asociados a los módulos que no están presentes en cada uno de ellos, es decir, las resistencias de pull-up para los sensores, las resistencias de pull-down para las líneas del teclado, el potenciómetro de regulación de contraste de la pantalla y el regulador de tensión con un capacitor en la entrada y en la salida funcionando como filtros de riple.

Además, como medida adicional, se agregaron cuatro borneras de conexión más, cada una con una resistencia de pull-up conectada a 5 V, con el sentido de facilitar el agregado, en una futura mejora del sistema, de pulsadores o sensores que puedan proveer una funcionalidad más al equipo.

Los conectores elegidos para cada uno de los componentes fueron los siguientes:

* Borneras a tornillo para los sensores infrarrojo e inductivo. Los mismos van montados fuera del gabinete principal y poseen cables de pocos milímetros de sección para su conexión.
* Un Jack para fuente de continua estándar y una bornera a tornillo como alternativa, por si la fuente comercial a utilizar no fuera compatible con el conector hembra de la placa.
* Conectores IDC macho para el teclado, la pantalla y el lector RFID. Los mismos proveen robustez en la conexión y prolijidad en la presentación, ya que los conectores hembra fueron montados en cable plano.
* Borneras a tornillo para la conexión futura de sensores o pulsadores ante una posible mejora.

### Fotos



Figura 5. Circuito impreso. Lado componentes. Montaje final.

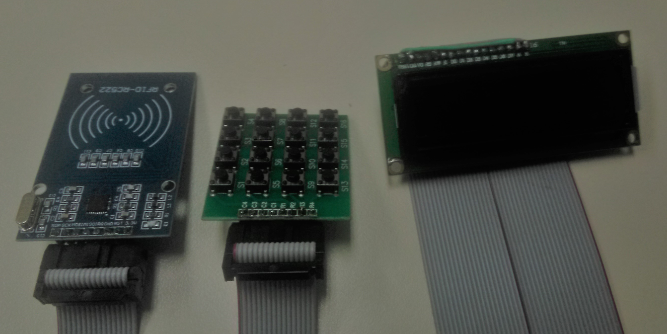


Figura 6. Módulos externos. Lector, teclado, display.

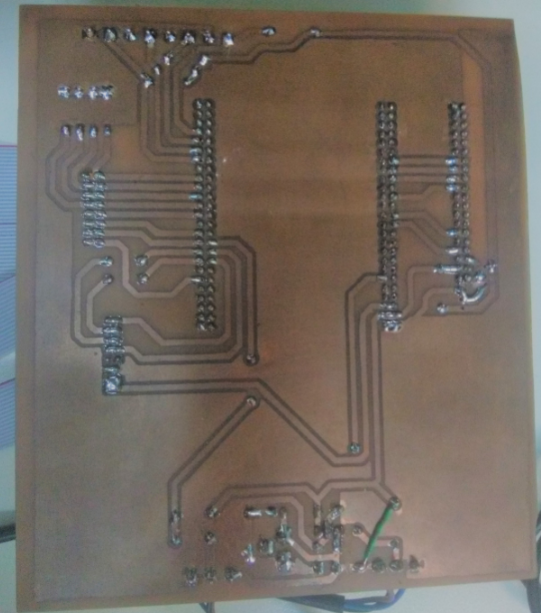


Figura 7. Circuito impreso. Lado soldadura. Finalizado.

## Software

### Enumeración de rutinas

* Rutina “main”: Contiene el comportamiento general del programa. Dentro de la misma hay dos secciones, una serie de llamadas a subrutinas que se ejecutan una única vez y luego un ciclo que se repite constantemente durante el funcionamiento del dispositivo.
* Subrutina “HAL\_Init”: Resetea todos los periféricos del microcontrolador, inicializa la memoria Flash y arranca el Systick.
* Subrutina “SystemClock\_Config”: Configura los registros relativos al Systick, habilita una interrupción cada un milisegundo.
* Subrutinas “MX\_GPIO\_Init, SPI1\_Init, TIM3\_Init, TIM4\_Init”: Inicializan los periféricos GPIO, SPI y TIMers según la configuración seleccionada en software CubeMX.
* Subrutina “MY\_FLASH\_SetSectorAddrs”: Define un sector de la memoria Flash para ser utilizado como memoria de almacenamiento no volátil del usuario.
* Subrutina “flash\_cargar\_operarios”: Lee la memoria Flash en el sector especificado previamente y carga los datos en un vector de estructura de datos personalizado.
* Subrutina “LCD\_Ini”: Inicializa el controlador del display según la hoja de datos.
* Subrutina “MFRC522\_Init”: Inicializa el controlador del módulo de lectura de tags RFID según la hoja de datos.
* Subrutina “display\_escribir”: Imprime dos cadenas de caracteres en la pantalla, una en cada línea.
* Subrutina “encuesta\_teclado”: Detecta la pulsación de una tecla del teclado matricial y almacena una bandera en un vector.
* Subrutina “conv\_hex”: Convierte los cinco bytes de identificación de cada tag en un vector con los correspondientes diez caracteres en ascii que los representan.
* Subrutina “MFRC522\_Check”: Detecta la proximidad de un tag RFID al lector y almacena su número de identificación.
* Subrutina “MFRC522\_Compare”: Devuelve un OK si los dos códigos de identificación que se le pasan como parámetro son iguales entre sí.
* Subrutina “flash\_guardar\_operarios”: Actualiza los datos guardados en la memoria que contienen la información sobre los operarios y sus respectivos trabajos.
* Subrutina de interrupción “HAL\_GPIO\_EXTI\_Callback”: La misma se ejecuta cuando alguno de los sensores detectó un objeto. Determina cuál de los sensores disparó la interrución.
* Subrutina “delayus\_block”: Delay bloqueante que dura los microsegundos especificados implementado con el timer 4. Se utiliza en la interfaz con el display, ya que los tiempos deben ser muy precisos, especialmente en la inicialización del mismo.
* Subrutina “introducir\_texto”: Controla el display y el teclado en conjunto para proveer un método de entrada de texto de 9 caracteres que se utilizarán para guardar el nombre de un operario.
* Subrutina “display\_unidades”: Muestra la cantidad de unidades detectadas del trabajo en curso, para ello tiene que convertir un dato numérico en caracteres ascii. Se utiliza la misma para mostrar la cantidad de golpes del trabajo en curso.

### Descripción del funcionamiento de cada rutina

#### Rutina “main”:

Como se observó anteriormente, la rutina main() tiene una sección que se ejecuta una única vez y otra que se ejecuta cíclica e indefinidamente. Se presentan los diagramas de estas dos secciones por separado para que sea más clara su representación.



Figura 8 - Diagrama de estados función main(). Ejecución única.

En el diagrama siguiente se observa la parte del código que se ejecuta continuamente durante el funcionamiento normal de la máquina. En las secciones siguientes se describirán las funciones y máquinas de estado contenidas en el diagrama anterior y el siguiente. Entre los cuales componen la totalidad del programa cargado en el microcontrolador.



Figura 9 - Diagrama de estados función main(). Ejecución cíclica.

#### Máquinas de estados y rutina sensor

En los diagramas siguientes, las flechas hacia arriba simbolizan que se presionó el botón “arriba”, las flechas hacia la izquierda que se presionó el botón “izquierda” y lo mismo para “derecha” y “abajo”.



Figura 10. Máquina de estados teclado (sin Menu\_4)



Figura 11. Máquina de estados teclado (Menu\_4)



Figura 12. Rutina sensor



Figura 13. Máquina de estados de tarjeta

Cabe destacar que, si el estado previo no es ninguno de los que figuran en el diagrama de estados anterior, la rutina es la siguiente:



Figura 14. Máquina de estados de tarjeta, continuación.

#### Subrutina “HAL\_Init”:

Esta función es provista por el juego de drivers HAL, al ejecutarse resetea todos los periféricos del microcontrolador, inicializa la memoria Flash y arranca el Systick.

#### Subrutina “SystemClock\_Config”:

Esta función, también provista por el conjunto de librerías HAL, configura los registros relativos al Systick, y habilita una interrupción cada un milisegundo.

#### Subrutinas “MX\_GPIO\_Init, SPI1\_Init, TIM4\_Init”:

Estas funciones son generadas por el software CubeMX y se encargan de configurar los periféricos GPIO, SPI1 y TIMer4 respectivamente según la configuración seleccionada en software CubeMX. Los parámetros seleccionados para cada uno de los periféricos mencionados fueron los siguientes:

Para los GPIO:

|  |  |  |
| --- | --- | --- |
| Pin | Modo | Denominación |
| PA1 | Output Push Pull | Teclado C4 |
| PA4 | Output Push Pull | RFID\_SDA |
| PB1 | Output Push Pull | Teclado C2 |
| PB5 | Output Push Pull | Display E |
| PB7 | Output Push Pull | Display RS |
| PC4 | External Interrupt Mode with Rising edge trigger detection | Sensor golpes |
| PC5 | Output Push Pull | Teclado C3 |
| PC11 | Output Push Pull | Display D7 |
| PD0 | Output Push Pull | Display D6 |
| PD2 | Output Push Pull | Display D5 |
| PD6 | Output Push Pull | Display D4 |
| PD12 | Output Push Pull | Led Verde |
| PD13 | Output Push Pull | Led Naranja |
| PD14 | Output Push Pull | Led Rojo |
| PD15 | Output Push Pull | Led Azul |
| PE6 | External Interrupt Mode with Rising edge trigger detection | Sensor unidades |
| PE7 | Output Push Pull | Teclado C1 |
| PE9 | Input mode | Teclado F4 |
| PE11 | Input mode | Teclado F3 |
| PE13 | Input mode | Teclado F2 |
| PE15 | Input mode | Teclado F1 |

SPI1:

|  |  |  |
| --- | --- | --- |
| Pin | Modo | Denominación |
| PA5 | SPI1\_SCK | RFID\_SCK |
| PA6 | SPI1\_MISO | RFID\_MISO |
| PA7 | SPI1\_MOSI | RFID\_MOSI |

TIM4: Se configuró como contador en forma ascendente, con un clock de 16 MHz y un valor de autoreload de 16. Proveyendo así una interrupción por reset cada 1 microsegundo.

#### Subrutina “LCD\_Ini”:

Al ejecutarse envía datos al driver de la pantalla LCD, según el siguiente diagrama, especificado en la hoja de datos del controlador.

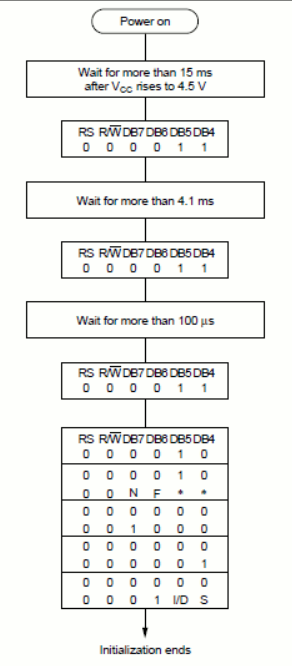


Figura 15 - Diagrama de flujo inicialización de pantalla.

#### Subrutina “MFRC522\_Init”:

Inicializa el controlador del módulo de lectura de tags RFID según la hoja de datos. Esta función está incluída en una librería opensource para el control del módulo elegido. Se intentó anteriormente realizar la configuración y comunicación con el controlador sin el uso de librerías externas, pero los resultados no fueron satisfactorios.

#### Subrutina “MY\_FLASH\_SetSectorAddrs”:

Define un sector de la memoria Flash para ser utilizado como memoria de almacenamiento no volátil del usuario. Es necesario enviarle como parámetro un número de identificación del sector y la posición de memoria en la cual inicia ese bloque. En el caso del proyecto se utilizó el sector 7, cuya dirección fue extraída de la hoja de datos del microcontrolador y es “0x 0806 0000”.

#### Subrutina “flash\_cargar\_operarios”:

Lee la memoria Flash en el sector especificado previamente y carga los datos en un vector de estructura de datos personalizado. El ciclo de lectura se realiza 20 veces, ya que está definido inicialmente que la cantidad máximo de operarios es 20. La variable operarios[], que es la variable de trabajo, es un vector de estructuras de datos llamadas Operario, compuesta por los campos:

condicion : 2 bytes.

nombre: 9 bytes.

id: 5 bytes.

golpes: 4 bytes.

unidades: 4 bytes.

productividad: 2 bytes.

Lo cual da un tamaño de estructura de 26 bytes. Se decidió dejar 6 posiciones libres a continuación de cada operario por si en algún momento fuera necesario agregar algún campo nuevo a cada uno, dejando un tamaño final de ocupación de espacio en memoria de 32 bytes por operario. A partir de estas consideraciones podemos definir un mapa de memoria como el que se observa a continuación. En el mismo las posiciones de memoria están representadas como el valor de desfasaje respecto de la posición inicial del sector 7 de la memoria Flash del microcontrolador.

A continuación, se esquematiza el ciclo de carga de los operarios en el vector de trabajo. Esto se realiza una única vez al comienzo del programa.



Figura 16 - Mapa de memoria de operarios.



Figura 17 - Diagrama de flujo flash\_cargar\_operarios()

#### Subrutina “display\_escribir”:

Imprime dos cadenas de caracteres en la pantalla, una en cada línea. Las líneas se le pasan a la función en forma de cadena de caracteres, es decir, con un carácter de finalización (0) según lo definido en el estándar C. Lo primero que hace la función es uniformar dicha cadena de caracteres a 16 (sin contar el carácter de fin de string), si el valor recibido tiene menos caracteres, rellena con espacios, si tiene más, trunca el contenido.

Una vez hecho esto, envía estas cadenas al display según lo indica la hoja de datos. La comunicación está implementada en la librería lcd\_txt, con una modificación en el código de la misma, para que utilice el TIMER4 para los intervalos de inicialización de la pantalla, ya que, si no, no funcionaba correctamente y había que reiniciar varias veces la placa hasta que el visor muestre correctamente los datos deseados.

#### Subrutina “encuesta\_teclado”:

Detecta la pulsación de una tecla del teclado matricial y almacena una bandera en un vector. Si se presionó una tecla, la variable f\_boton pasa al valor 1 y, por consiguiente, el ciclo principal entra a la “máquina de estados teclado” para resolver la instrucción. Si se presiona más de una tecla simultáneamente se almacena solo una.



Figura 18 - Diagrama de flujo encuesta\_teclado

#### Subrutina “conv\_hex”:

Convierte los cinco bytes de identificación de cada tag en un vector con los correspondientes diez caracteres en ascii que los representan. Para ello, mediante un ciclo for recorre el vector de destino. El ciclo guarda la parte menos significativa (4 bits = 1 hexadecimal) de cada posición del vector de origen en el correspondiente orden del vector de destino. Inmediatemente después, mediante un condicional switch se reemplaza el valor de cada conjunto de 4 bits por el valor en ascii que lo represente en hexadecimal, para obtener una visualización más agradable, dado que sin esta conversión, los 5 caracteres id son totalmente aleatorios.



Figura 19. Ejemplo de conversión

#### Subrutina “MFRC522\_Check”:

Es una función incluída en una librería externa que detecta la proximidad de un tag RFID al lector y almacena su número de identificación. La comunicación con el módulo es muy compleja, por eso no fue implementada “a mano”, pero a grandes rasgos envía comandos hacia el chip MFRC522, vía SPI, le pregunta si hay una tarjeta en el área de detección, y si es así, envía un comando que devuelve el id de dicha tarjeta. El mismo tiene 5 bytes que se almacenan en un buffer que se le pasa a la función como parámetro. La función retorna el estado de la operación, puede ser OK, ERROR, COLISION, entre otros. Sólo me interesa el OK, que sólo es retornado si se leyó una tarjeta y si la lectura fue correcta.

#### Subrutina “MFRC522\_Compare”:

Es sólo un ciclo for que compara dos vectores, devuelve un OK si los dos vectores que se le pasan como parámetro son iguales entre sí. Se utiliza para chequear si el ID de la tarjeta leída está asociado a algún operario cargado en la base de datos o no.

#### Subrutina “flash\_guardar\_operarios”:

Genera un vector de 640 bytes, que es el total de memoria que se utiliza para la base de datos de operarios. Cíclicamente va cargando en este vector los datos de los operarios con las modificaciones que haya sobre la estructura de trabajo operarios[ ] respetando el mapa de memoria mostrado en la Figura 16. Una vez relleno este vector, se escribe mediante una función de escritura de la memoria flash de una librería externa (MY\_FLASH\_WriteN()) el vector completo. Para hacer esto, por el propio funcionamiento de la memoria flash es necesario el borrado del sector completo previamente a su escritura. Por eso, es necesario cargar todos los operarios juntos y no solo actualizar las modificaciones.

#### Subrutina de interrupción “HAL\_GPIO\_EXTI\_Callback”:

Es una subrutina que tiene lugar sólo si alguno de los sensores (inductivo, o infrarrojo) generó un pulso, o sea, una detección. Las instrucciones que tienen lugar en una función de callback deben ser de muy rápida ejecución, por lo tanto, en este punto sólo se levantan banderas. Si el pin de interrupción que la generó fue el de unidades, se pone en uno un flag que indica que hubo una detección y un flag que indica que fue el sensor de unidades. Si el pin de interrupción que la generó fue el de unidades, se pone en uno el flag que indica que hubo una detección y se pone en cero el flag que indica que fue el sensor de unidades.

#### Subrutina “delayus\_block”:

Para lograr un delay bloqueante de tempo variable en microsegundos se utiliza un timer configurado previamente para que aumente su contador a 16 MHz, esto implica un período de 1/16 microsegundos. Al llamar a esta función se pone un flag en cero, se carga en el registro de comparación del timer el entero que se le haya pasado como parámetro multiplicado por 16, se arranca el timer y a continuación entra en un ciclo while que chequea el estado del flag que se puso en cero. Por otro lado, en la función de callback por CTC del mismo timer, se pone el flag en 1, por lo tanto, apenas salte la interrupción, la condición del while será falsa, y el programa retornará a su ejecución secuencial.

#### Subrutina “introducir\_texto”:



Figura 20. Diagrama de flujo introducir\_texto

#### Subrutina “display\_unidades”:

Muestra en la pantalla la cantidad de unidades del trabajo en curso, para ello tiene que convertir un dato numérico en caracteres ascii. Se utiliza la función sprintf() para poder imprimir el valor entero en una variable tipo cadena de caracteres.

# Modo de operación

## Inicio del equipo

El equipo se enciende con la llave ubicada en el lateral izquierdo del gabinete, siempre y cuando haya sido conectada previamente la fuente de alimentación cuyo conector se encuentra ubicado en el mismo lateral. El dispositivo está listo para operar cuando se observa en la pantalla la leyenda “Inicio de la máquina”.

## Teclado

El equipo se comanda mediante un teclado de 16 botones de los cuales las tres primeras columnas se reparten las funciones señaladas en el siguiente gráfico. A su vez, el estado actual del dispositivo se muestra al usuario en la pantalla y pretende, mediante mensajes, hacer más intuitivo el uso del dispositivo.



Figura 21. Distribución teclado.

Las flechas izquierda y derecha permiten desplazarse entre distintos menús que se encuentren al mismo nivel jerárquico, la flecha hacia abajo permite descender un nivel, es decir, acceder a un submenú del menú actual, de manera contraria, la flecha hacia arriba vuelve al menú del cual forma parte el submenú actual. El botón Enter actúa como OK, en caso de estar en un menú, realiza la misma función que la flecha hacia abajo y en caso de estar en una función, la ejecuta. El botón Esc vuelve al Menú 0 “Inicio” desde cualquier instancia del menú.

## Menús

A continuación, se muestran los distintos menús mediante un diagrama de árbol, el sentido de la flecha indica el botón que hay que presionar para pasar de un menú a otro, es decir, si la flecha de A a B apunta hacia la derecha, el botón que lleva a B desde A es el botón “derecha”.



Figura 22. Árbol completo de menús.

## Operación

El equipo cuenta con una base de datos de hasta 20 operarios de los cuales el primero ya está cargado. Para comenzar, cargue los operarios que desee dirigiéndose mediante el teclado hasta el menú [4. Configuración], submenú [4.4. Alta operario] y siga las instrucciones en la pantalla. La operación solicitará que confirme su identidad mediante el acercamiento del tag maestro.

Una vez que tenga registrados a sus operarios, con la máquina en cualquier menú, al acercar un tag que esté asociado a un operario cargado previamente, se iniciará un nuevo trabajo. A partir de ese momento, los contadores de golpes y unidades comenzarán a ser mostrados en la pantalla y serán asociados al operario que acercó su tarjeta al lector. Para finalizar el trabajo, sólo debe acercar su tarjeta nuevamente y se guardará automáticamente el registro de golpes y unidades registradas. Si se acerca una tarjeta que no coincide con la del operario que inició el trabajo, no tendrá ningún efecto.

Si no hay trabajos en curso y desea ver algún registro puede hacerlo desde el teclado, navegando en el menú como se muestra en el diagrama de la Figura 22. Puede observar los operarios registrados, los totales generales, y la productividad de cada uno de los empleados (si presiona la tecla “abajo” en el submenú productividad, se muestra el número de unidades en base a la que la misma fue calculada).

Desde el menú de configuración puede dar de baja operarios cargados, (esto se verá reflejado en los totales, ya que a partir de que se borren los operarios, lo realizado previamente no será tenido en cuenta), reseteo de totales, habilitación de categoría “Maestro” para otros operarios (pensado para encargados o jefes de producción) y dar de alta nuevos operarios.

# Ensayos

Se comprobó que el efecto de todos los comandos de teclado en cada una de las instancias del menú sea el indicado en los diagramas previos, con resultados satisfactorios.

También se comprobó que los códigos de identificación de RFID son guardados y procesados correctamente. Eso además verifica la no volatilidad de los datos, gracias a que son almacenados y leídos desde la memoria flash.

Además se verificó que los sensores de cuenta no tomen valores erróneos, mediante la repetición de ciclos de sensado de prueba, lo cual también arrojó resultados satisfactorios ya que no se detectó ninguna anomalía en las cantidades de “veces” informadas en el display.

# Conclusiones

En lo personal estoy contento con el resultado del proyecto.

Objetivamente, el grado de cumplimiento de los objetivos planteados al incio del proyecto es prácticamente completo. El único punto no cumplido fue el agregado de un display contador “gigante” que decidió darse de baja porque la información iba a ser redundante, ya que el display LCD mostraría lo mismo y más, y porque el potencial usuario del dispositivo final considera que podría ser perjudicial para los empleados el hecho de que información tan llamativa esté cerca del área de trabajo, ya que eso podría generar una distracción que podría llevar a un accidente laboral.

Considero que la interfaz lograda es suficientemente intuitiva, los métodos de entrada de datos responden de manera satisfactoria, tanto el teclado, como el lector de tarjetas. La información mostrada en la pantalla es significativa. Las funciones ideadas originalmente fueron implementadas en su totalidad y, además, se dejó implementada en el circuito impreso un conjunto de cuatro entradas con sus respectivas resistencias de pull down listas para conectar a cualquier otra llave o pulsador que pudiera ser necesario para una ampliación a futuro.

Lo más complicado del proyecto resultó la puesta en funcionamiento del lector de RFID, si bien la interfaz SPI no es compleja de utilizar, el integrado que controla al dispositivo tiene una manera rebuscada de inicializar y operar, a tal punto que debí utilizar una librería ajena para manejarlo, cosa que no fue sencilla de encontrar. Fuera de eso, leyendo las hojas de datos de la placa STM32F411 y de los distintos sensores utilizados, se encontró la solución a todos los imprevistos de funcionamiento encontrados.

El módulo pensado en un principio para leer las tarjetas RFID, el RDM6300, que posee interfaz uart y sólo sirve para leer el código de identificación de los tags (a diferencia del MFRC522 que lee el ID, lee la memoria, escribe datos en la tarjeta, etc) no se pudo utilizar debido a que no funcionó ninguno de los ejemplares que adquirí. Nunca me había pasado con otros módulos del estilo, así que personalmente, no recomiendo este módulo para trabajar en ningún proyecto en adelante.

Me fue muy útil para el desarrollo del trabajo la elaboración de una planilla de cálculo donde figuran todos los pines que utilizo de la placa, a qué va conectado cada uno y cómo está configurado el pin. Además, fue útil diseñar un mapa de memoria para el almacenamiento de las variables no volátiles y la creación de una grilla tamaño display para diseñar mejor los mensajes en pantalla.

Por último, hubo un antes y un después radical al implementar el PCB de manera robusta y afirmando los contactos. Durante las primeras pruebas, especialmente con el display conectado con pines macho-hembra clásicos, se generaban errores aleatorio inexplicables que se debían a falsos contactos o movimientos en los cables. El tiempo de desarrollo bajó muchísimo una vez que se implementaron bien las conexiones. La utilización de cable plano y conectores IDC dio resultados excelentes para eso.

# Proyecto finalizado

En esta sección se deberá colocar una foto “cuidada” del producto final para dar un cierre al informe. Al decir “cuidada” se hace referencia a una foto de buena calidad, con fondo liso (en general blanco) y con un ángulo que permita apreciarlo bien.

El tamaño de esta imagen podrá ocupar toda una página y ser colocada en forma horizontal si es necesario.

# Referencias

1. UM1842 – User Manual - Discovery kit with STM32F411VE MCU – Rev1, Diciembre 2014, ST
2. MB1115 – Schematic diagram Discovery kit with STM32F411VE MCU – Rev B.2,Octubre 2014, ST
3. MFRC522 Standard performance MIFARE and NTAG frontend – Rev 3.9, Abril 2016, NXP Semiconductors
4. RDM630 Specification
5. El Lenguaje de Programación C – Segunda edición,1991, Kernighan y Ritchie, Pearson Education.
6. MYaqoobEmbedded Youtube channel – Disponible en: <https://www.youtube.com/channel/UC-CuJ6qKst9-8Z-EXjoYK3Q>

**Universidad Nacional de La Matanza**

**Técnicas Digitales III**