



Laboratorio de Microcomputadoras - 66.09

S.H.I.V.A

Sistema Híbrido Inclinador de Vaso Automatizado

Profesor:	Ing. Guillermo Campiglio								
Cuatrimestre/Año:	1º/2019								
Turno de las clases prácticas	Miércoles								
Jefe de trabajos prácticos:	Ing. Ricardo Arias								
Docente guía:	Ing. Fabricio Baglivo								
Autores									
Nombre	Apellido	Padrón							
Lola	Segura	99254							
Martín	Quattrone	98644							

Observaciones:

.....
.....
.....
.....
.....
.....
.....
.....
.....
.....
.....

Fecha de aprobación		Firma J.T.P

Coloquio
Nota final
Firma profesor

Índice

1. Introducción	2
2. Descripción del proyecto	2
3. Módulos del proyecto	2
3.1. Celda de carga	3
3.2. Servo motor	4
3.3. Pantalla LCD	4
3.4. Integración de los módulos al micro	5
4. Software	6
5. Resultados	11
6. Conclusiones	12
7. Bibliografia	13
8. Apéndice	13
8.1. Gastos	13
8.2. Hojas de datos	13

1. Introducción

Se realizó un sistema para automatizar la inclinación de un vaso de cerveza a medida que es servida para evitar la formación de espuma excesiva en esta.

Para ello se utilizó como parámetro el peso del líquido, a partir del cual el dispositivo toma distintos ángulos de inclinación.

En el presente trabajo se explicará cómo funciona el dispositivo y que consideraciones se tuvieron en cuenta para realizarlo, comentando por separado cada una de las partes que lo conforman.

2. Descripción del proyecto

El proyecto consta de una plataforma, sobre la cual se apoya el vaso, que rota sobre un eje horizontal logrando así tomar los distintos ángulos de inclinación necesarios. La plataforma es sostenida desde su parte superior por dos ménsulas por medio de dos tornillos que conforman su eje de giro. Este sistema está montado sobre una base de madera. El resultado final se pueden ver en la Figura 5.1.

Para inclinar la plataforma se utiliza un servo motor que tira de la parte inferior con un hilo. El servo lleva a la plataforma a la posición de mayor inclinación, y una vez que se llega a servir la mitad del líquido, a medida que se sigue vertiendo cerveza, la plataforma se va declinando hasta alcanzar nuevamente la posición inicial, esta secuencia se puede observar en la Figura 5.4.

Se utilizó como parámetro el peso del líquido, para lo que se trabajó con una celda de carga sobre la cual se colocó toda la estructura. Mediante la lectura constante del peso se logró utilizar este parámetro no sólo para definir el ángulo de inclinación de la plataforma sino toda la configuración del dispositivo. Mediante una pantalla LCD (ejemplo en la Figura 4.1) se guía al usuario para que, realizando dos golpes sobre la base inicie el proceso, elija la medida de cerveza que desee y, en caso de ser necesario, cancele la operación, siendo todo esto pudiendo ser controlado con el peso leído.

3. Módulos del proyecto

Se presenta a continuación un diagrama de bloques del proyecto, donde se pueden observar todos los elementos utilizados. En esta sección se describirá el funcionamiento de cada uno de ellos.

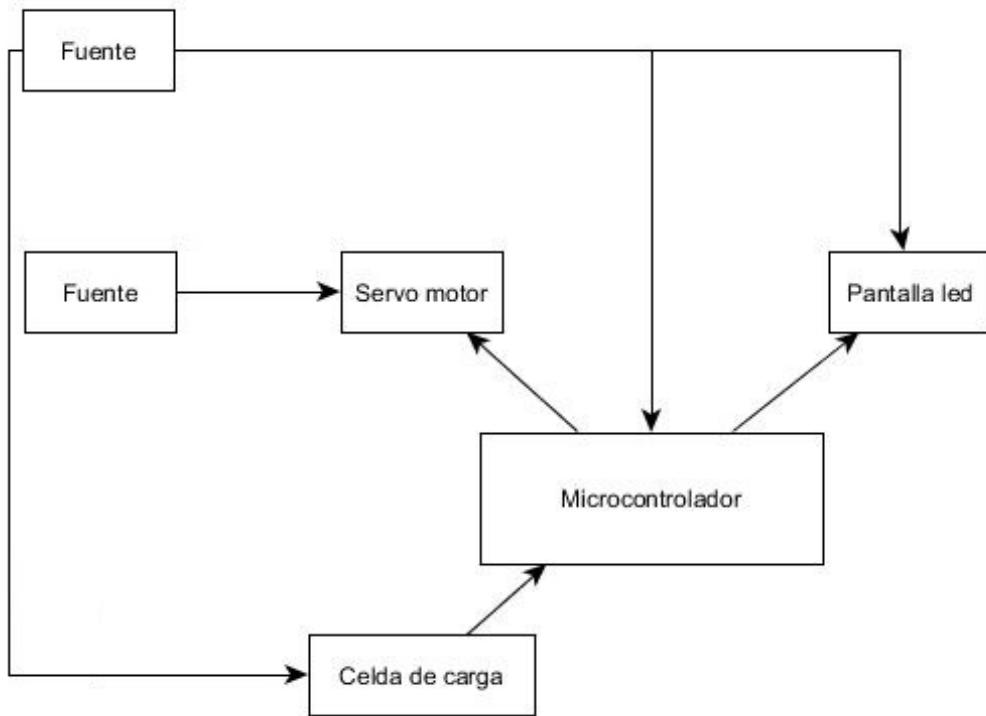


Figura 3.1: Diagrama de conexiones en bloque

3.1. Celda de carga

Una celda de carga es un dispositivo que convierte fuerza en una señal eléctrica. Para conectar la celda con el microcontrolador se utiliza un módulo amplificador Hx711 que convierte la lectura analógica en digital, generando datos de 24 bits.

El amplificador consta de dos pines (*PD_SCK* y *DOUT*) para controlar la entrega de los datos. El pin *DOUT* indica cuando el amplificador tiene datos para entregar, mientras se mantenga en alto no se pueden leer datos. Cuando cambia, para obtener los datos se debe enviar una señal de *clock* de 25 pulsos¹ por el pin *PD_SCK*, por cada uno de ellos se envía un *bit* (empezando por el más significativo) desde el amplificador al microcontrolador. El último pulso se utiliza para volver a *setear* el pin *DOUT* y aguardar por un nuevo dato.



Figura 3.2

Para este proyecto se trabajó con una celda de 10 kg de peso máximo montada entre dos maderas utilizando separadores como se puede ver en la Figura 3.2. Sobre la madera superior se montó toda la estructura del proyecto.

¹El numero de pulsos define el valor de ganancia, con 25 pulsos se tiene una ganancia de 128.

3.2. Servo motor

Un servo motor es un motor que permite controlar la posición de su eje a través de una señal PWM (modulación por ancho de pulsos), por lo que, a diferencia de un motor de DC, se mueve una cierta cantidad de grados y puede mantener una posición fija.

El servo motor utilizado es de rango de giro limitado, es decir, posee una rotación de 180° . Para manejarlo se trabaja con ondas cuadradas de periodo $T = 20\text{ ms}$. Dependiendo del ancho del pulso el motor adopta una posición fija. Como puede verse en la Figura 3.3, el pulso puede durar desde $0,5\text{ ms}$, correspondiéndose a la posición -90° , hasta $2,5\text{ ms}$, correspondiéndose a la posición 90° .

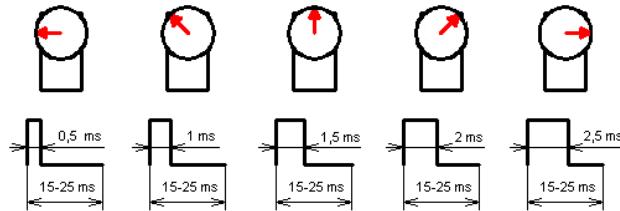


Figura 3.3

La fuerza que puede hacer depende de la tensión entregada, y se mide en kilogramos por centímetro, cuanto mas alejado está el punto de trabajo del servo con respecto al eje menos es la fuerza que puede ejercer. En este caso se trabajó con un servo que podía ejercer a una fuerza máxima de $6,5\text{ kg} - \text{cm}$ si le entregaban 6 V de tensión.

3.3. Pantalla LCD

Se trabajó con una pantalla LCD (cristal líquido) de 16×2 , es decir, tiene dos filas de 16 caracteres cada una. La misma posee además de los pines de alimentación, otro para ajustar el contraste, los cuales se indican en la Figura 3.4. Por otro lado, posee 8 pines para el envío/lectura de caracteres. Sin embargo, se pueden enviar datos utilizando solo 4 de estos bits (D7:D4), enviando cada byte de información en dos *nibbles*, primero el más significativo y luego el menos significativo.

Por último, la pantalla cuenta con tres pines que sirven para configurar el comportamiento de la misma (*RS*, *RW* y *E*). El pin *RS* indica: si esta *seteado*, que se va a mandar un dato, y si esta en cero, que se va a mandar una configuración para la pantalla. Por otro lado, con el pin *RW* se le indica a la pantalla si va a funcionar en modo de lectura o en modo de transmisión. Por ultimo, el pin *E* es utilizado para enviar pulsos y obtener/mandar así, los bytes de datos o comandos.

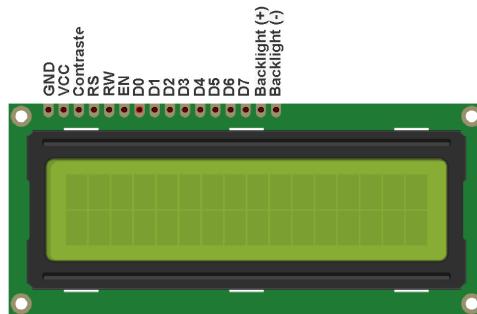


Figura 3.4

3.4. Integración de los módulos al micro

A continuación se presenta una lista de los componentes utilizados en el proyecto. Además, en la Figura 3.5 se observa el esquemático, donde se presentan las conexiones entre los módulos mencionados y el microcontrolador.

- Microcontrolador:
 - Atmega 328p
- Celda de carga:
 - Celda de carga de capacidad nominal 10 kg
 - Amplificador Hx711
- Servo motor:
 - SG5010 micro servo
 - Integrado LM7806² - Entrega 6 V al conectarle una fuente de 7-18 V
- Pantalla LCD:
 - Display LCD 16 caracteres 2 líneas Controlador HD44780
 - Resistor variable multivuelta de 3 terminales - Para regular el contraste de la pantalla
- Fuente:
 - Batería de 9 V alcalina x2
 - Interruptor Deslizante Botón Switch Encendido - Apagado

²Hoja de datos: <https://datasheetspdf.com/pdf/918901/FairchildSemiconductor/LM7806/1>

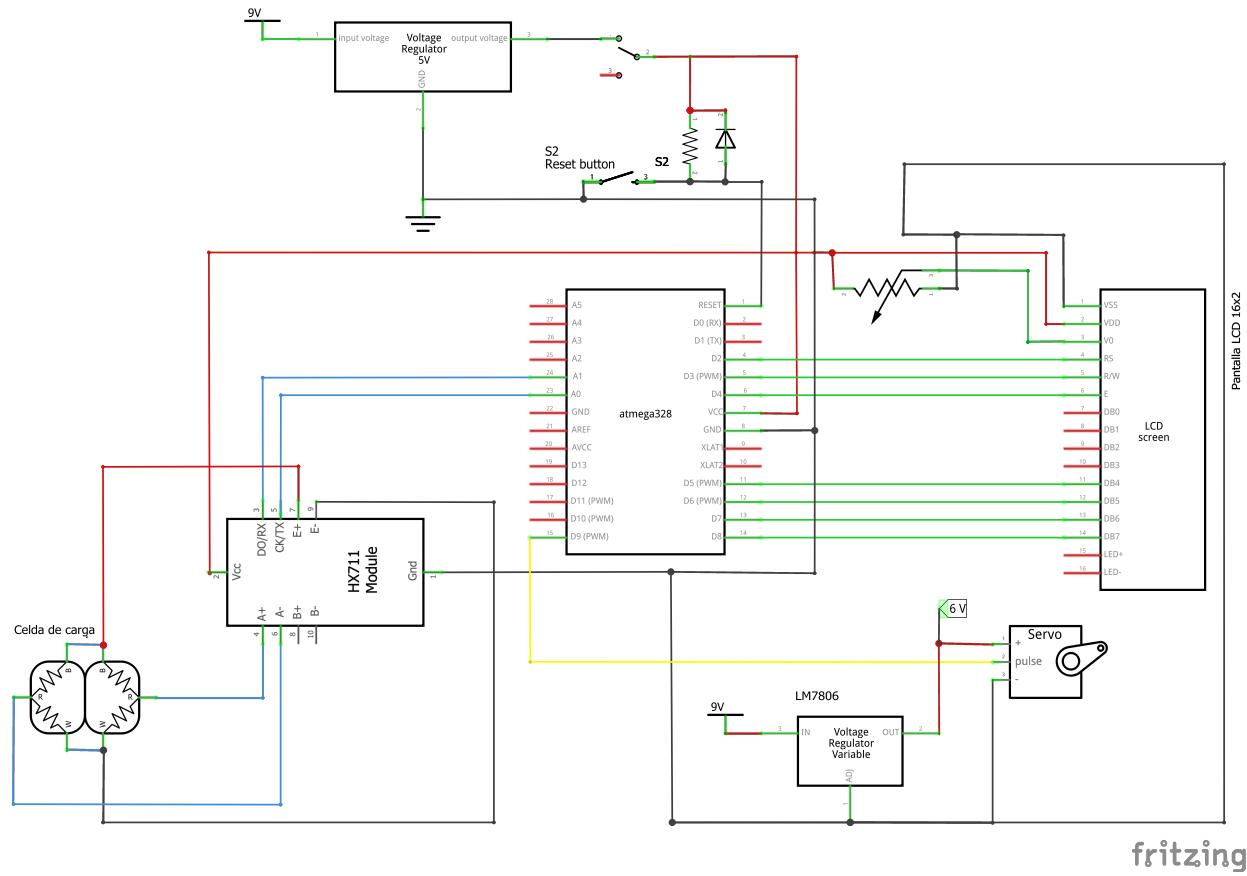


Figura 3.5: Esquemático

4. Software

Primero se realizaron las configuraciones pertinentes para trabajar con la celda de carga ya que el resto del trabajo se debía realizar en función del peso. Como se explicó en la sección previa, el amplificador de la celda de carga entrega datos de 24 bits, los mismos no tienen unidades de peso, sino que son valores aleatorios que dependen de la celda trabajada. Por lo que en primer lugar se obtuvo el valor que leía la celda cuando no tenía ningún objeto colocado sobre ella, y este era el valor que había que restarle a cada lectura, el valor de *tara*.

Luego se debió encontrar un factor de escala mediante el cual se pudieran transformar los datos leídos a unidades de peso. Para esto, se buscó un objeto cuyo peso fuera conocido con certeza y que sea del orden del peso máximo de la celda. A partir del valor leído para ese objeto se obtuvo el factor de escala por el que había que dividir cada dato para convertirlo a gramos mediante la ecuación

$$\text{FACTOR} = \frac{\text{VALOR DE LECTURA}}{\text{PESO REAL}} \quad (4.1)$$

Una vez obtenido este valor, se crearon las funciones que luego de leer cada dato que entrega el amplificador, le resta la *TARA* y lo divide por el *FACTOR* encontrado experimentalmente. Para obtener un valor más preciso, la *TARA* se calculó a partir del promedio de 16 muestras.

Hasta este punto para observar las lecturas de la celda se trabajó con la *UART* (*Universal Asynchronous Receiver-Transmitter*) en modo de recepción. Se tuvo que tener cuidado a la hora de realizar esto ya que, con este método se reciben datos de 8 bits y los datos entregados por el amplificador eran de 24 bits.

Habiendo calibrado la celda, se procedió a crear las funciones que trabajan con el peso leído. En primer lugar,

la función que inicializa el programa, la cual detecta cuando el usuario realiza dos golpes sobre la estructura. Para realizarla, se midió experimentalmente que lecturas realizaba la celda al golpearla moderadamente con la mano. Se obtuvo que un golpe se veía reflejado como valores que rondaban los 300 grs aproximadamente, por lo que la forma de armar la función fue la siguiente: aguardar a leer un peso mayor a 255 grs, al leer uno, chequear si en alguno de los próximos 5 datos hubo un peso menor a 255 grs, al encontrar uno que sí, chequear si en los próximos 5 datos se leía un dato mayor a 255 grs, lo cual representaría el segundo golpe.

Iniciado el proceso, se aguarda a que el usuario coloque un vaso en la plataforma. Para ello, se creó la función que aguarda a leer 5 muestras iguales mayores a 200 grs, que es un valor encontrado experimentalmente para el peso mínimo de una vaso de vidrio. No se pudo lograr que el proyecto sea válido para vasos de plástico debido a que estos rondan los 15 – 30 grs, por lo que al apoyar objetos livianos sobre la estructura equivaldría a que el programa interprete que corresponden a un vaso.

Para cancelar el proceso se decidió que el usuario debe realizar presión sobre la estructura. Nuevamente, mediante mediciones experimentales se obtuvo a cuánto peso equivalía realizar una presión moderada, obteniendo valores mayores a 4000 grs. Por lo tanto, luego de leer cada peso, mediante una función se chequea si este valor leído es mayor a 4096 grs y en caso que lo sea, se finaliza el programa volviendo a su inicio.

Una vez que se detectó un vaso, se llama a la función que *setea* la *TARA* para que las próximas lecturas del peso equivalgan al líquido que se vuelca en el vaso. Pero antes el usuario debe elegir la medida de cerveza que va a servir, dentro de las 3 opciones disponibles, *PINTA*, *CHOPP* o *MEDIA-PINTA*.

Estas medidas ya tienen configurado la cantidad de mililitros que representan y para ir iterando entre las opciones, el usuario debe golpear la estructura de la misma forma que lo hizo para iniciar el programa. Por defecto la medida elegida inicialmente es *PINTA*, y si al pasar 4 segundos no se detectaron 2 golpes, es la medida que queda seleccionada. Si ocurre una perturbación, pasa a la siguiente medida y se reinician los 4 segundos que posee el usuario para pasar a otra. Una vez elegida, el valor de la misma se guarda en dos registros y se pasa a la siguiente función. Para realizar todo esto, se implementó el *timer* de 16-bits que posee el microcontrolador *Atmega328p*. Cada vez que se cambia de medida, se inicializa este *timer*, y al pasar los 4 segundos, salta la interrupción que hace que el programa pase a la siguiente función.

En este punto, se llama a la función que declina el vaso en función del líquido servido por el usuario. Previo a eso, se lleva la plataforma a la posición de máxima inclinación al cambiar la posición en la que se encuentra el *servo*. Las posiciones de este son controladas a través del *Timer0* en modo *Fast-PWM* y haciendo uso del pin asociado a él. De esta forma, se observó que el *servo* posee 32 posiciones que cubren el rango de ángulos que puede girar.

Con el vaso totalmente inclinado, la forma de estructurar la función fue la siguiente: se toma el valor de la *MEDIDA* seleccionada y se lo divide por 64, obteniendo un *PASO*. Este valor se utiliza para definir inicialmente un *TECHO*, y cuando se lee un peso mayor a él, este se redefine como *TECHO=TECHO+PASO* y se vuelve a leer peso. La declinación comienza en la segunda mitad de la *MEDIDA* por lo que si se lee un peso mayor al *TECHO* y si además, este valor es mayor a la mitad de la *MEDIDA*, se mueve el *servo* una posición. Dado que $PASO = \frac{MEDIDA}{64}$, en la segunda mitad se redefine el *TECHO* 32 veces, por lo que se recorren las 32 posiciones que posee el *servo*, de ahí la elección del valor del *PASO*.

Para guiar al usuario mientras sirve la bebida, en la pantalla se puede observar una barra de progreso, ilustrada en la Figura 4.1. Para lograr esto, se envía un carácter negro³ a la pantalla que se agrega a los que ya se enviaron. Dado que una línea de la pantalla tiene espacio para 16 caracteres, este carácter se envía cada 4 redefiniciones del *TECHO*, llenando la línea luego de 64 redefiniciones que es cuando se llega al peso de la *MEDIDA*.

³valor ASCII: 0xFF

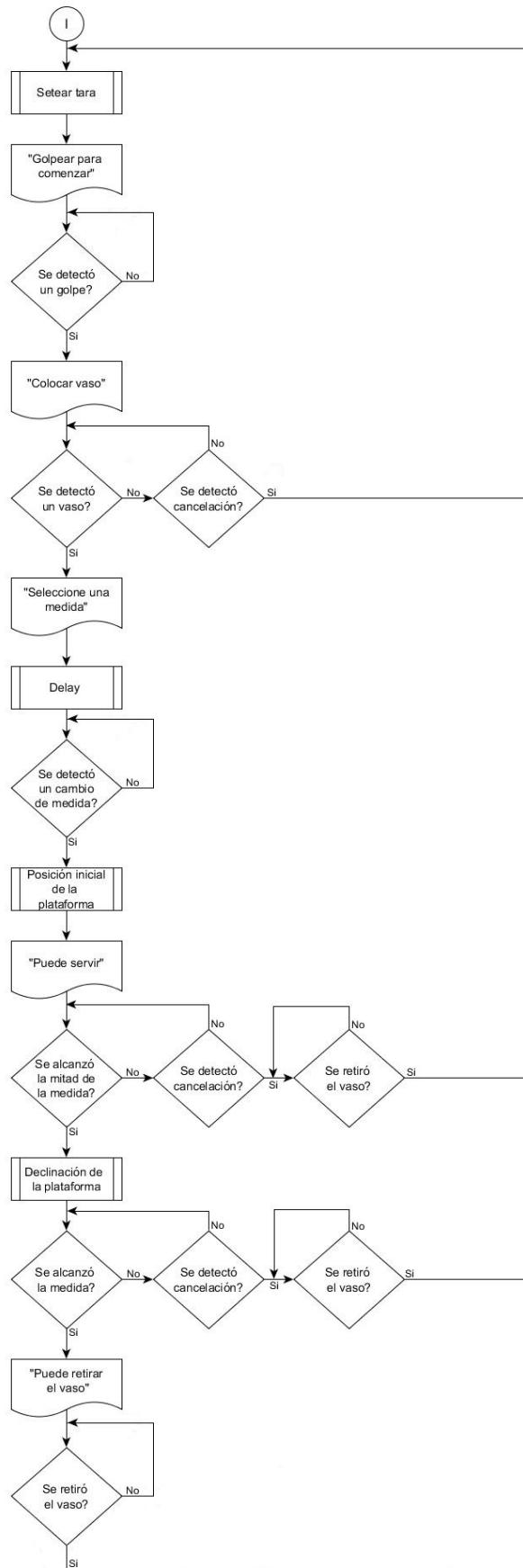


Figura 4.1: Proceso

Todo el proceso es acompañado por mensajes presentados en la pantalla LCD, que van guiando al usuario. Para esto se almacenaron en memoria *ROM* los mensajes que serían utilizados y se accedió a ellos mediante el puntero *Z*. Por lo que, para enviarlos simplemente se inicializa el puntero con la dirección en memoria que correspondiera y se llama a una función que se encargara de enviar el dato por la pantalla. Para simplificar aún más este proceso, se crearon *macros* para cada una de las direcciones utilizadas.

Dentro de las consideraciones evaluadas para el manejo de los mensajes vale la pena mencionar, en primer lugar, que para los textos que ocuparan más de 16 caracteres, se decidió tener un carácter ('\n') asignado exclusivamente para determinar cuando se debía pasar a la segunda línea de la pantalla. Además, se reservó otro carácter (0xF0) para indicar el final del mensaje. Por lo tanto, la función envía carácter por carácter, pero primero verifica que no sea ninguno de los comandos reservados.

Por último, a la hora de elegir las medidas se decidió incluir en la memoria ROM, a continuación de cada uno de los mensajes que indicaba la opción, el contenido en mililitros correspondiente a esta. Por lo que, se realizó una función que, al recorrer las opciones, primero se enviaba el mensaje y luego incrementaba el puntero para guardar en registros reservados el valor de la medida. Además, para lograr un menú de opciones circular, se agregó el carácter de fin de mensaje luego del contenido en ml de la última opción indicando así que se debía inicializar el puntero nuevamente con la primer opción.



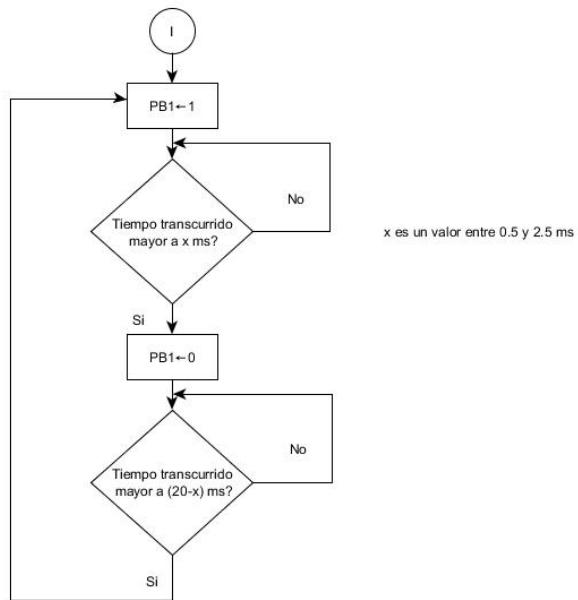


Figura 4.2: Diagrama de Flujo de la generación del PWM

5. Resultados

Los resultados del proyecto terminado se muestran en las siguientes imágenes.



Figura 5.1: Imagen del proyecto terminado



Figura 5.2: Cableado del proyecto

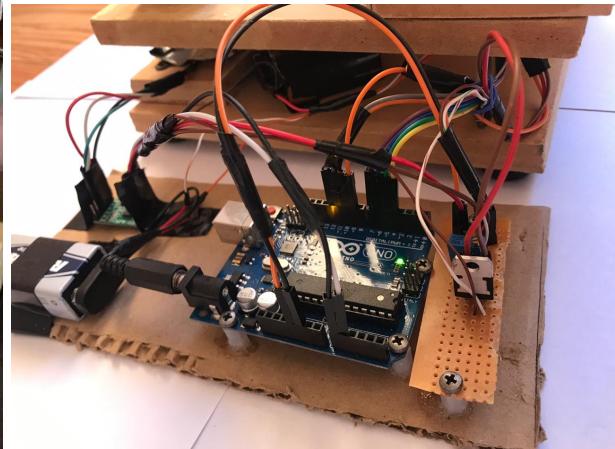


Figura 5.3: Cableado del proyecto

A continuación se muestran unas imágenes tomadas del proyecto funcionando en la sección de declinación del vaso a medida que se sirve cerveza.



Figura 5.4: Imágenes del proceso de servido del proyecto terminado.

6. Conclusiones

A modo de cierre, se analizarán los resultados obtenidos y se presentaran mejoras que se podrían incluir en un futuro al proyecto.

El proyecto cumplió con las expectativas planteadas en la etapa de planificación. Se logró un buen manejo del peso no solo para la lectura del líquido sino también para la configuración del programa a través del usuario, evitando la implementación de botones u otra forma de acceso. Por ejemplo, la inicialización del programa es a partir de dos golpes moderados, cualquier otro tipo de perturbación no arrancará el programa, por lo que el sistema es eficaz.

Por otro lado, la manera de inclinar la plataforma permitió el movimiento de objetos relativamente pesados (del orden de medio kilo) para la dimensión del proyecto, sin la necesidad de realizar mucha fuerza.

Por último, vale mencionar que, por la forma de implementar el manejo de selección de las medidas explicado en la sección 4 sería sencillo agregar más opciones. Simplemente se debería agregar el mensaje y el valor de la misma a continuación de las ya existentes y, luego de cargarlo en el microcontrolador, quedaría listo para su uso con la nueva medida ya implementada.

Sin embargo, existen ciertas mejoras que se podrían implementar en el proyecto y que serán presentadas a continuación. En una primera instancia se había planificado que el dispositivo pudiera ser utilizado con cualquier tipo de vaso. El problema que se halló al tratar de realizar dicha configuración fue que, al ser tan livianos los vasos de plástico, teniendo un peso del orden de los 10 grs, la función para detectar un vaso iba a tomar cualquier objeto que se apoyara sobre la estructura como uno.

Por otro lado, en cuanto al diseño, la idea era acomodar la estructura de forma tal que todos los componentes electrónicos quedaran entre las dos tapas de madera para reducir el tamaño del objeto y que además no quedaran a la vista. Desgraciadamente no quedó el suficiente espacio, por lo que se tuvieron que acomodar al costado de la estructura.

Por último, la idea principal del proyecto era realizar, además de la estructura que inclinara el vaso, una canilla de forma tal que no se tuviera que servir de manera manual la bebida. Esto no se implementó ya que no era viable realizar todo en el tiempo en el que se tenía para realizar el trabajo.

De todas formas, el proyecto será continuado implementando una válvula solenoide que se abra y cierre electrónicamente, integrándola a la función de inclinación del servo para controlar su funcionamiento por software. Al agregar una canilla que sirva la bebida de manera automática, el usuario se podrá independizar de este proceso y deberá encargarse únicamente de la configuración del dispositivo.

7. Bibliografía

- Celda de Carga.
 - <https://electronilab.co/wp-content/uploads/2017/11/HX711.pdf>.
 - https://naylampmechatronics.com/blog/25_Tutorial-trasmisor-de-celda-de-carga-HX711-Ba.html.
 - <https://github.com/bogde/HX711>
- Servo motor
 - <http://panamahitek.com/que-es-y-como-funciona-un-servomotor/>
- Pantalla LCD
 - Muhammad Ali Mazidi, Sarmad Naimi, Sepehr Naimi, (2011), *The AVR microcontroller and embedded system. Using assembly and c*, Upper Saddle River, New Jersey: Pearson Education.

8. Apéndice

8.1. Gastos

- Display LCD: \$150
- Servo motor: \$300
- Celda de carga: \$400
- Arduino Uno: \$550
- Plataforma de acrílico: \$800
- Otros: \$550

Generando un gasto total de \$2750.

8.2. Hojas de datos



24-Bit Analog-to-Digital Converter (ADC) for Weigh Scales

DESCRIPTION

Based on Avia Semiconductor's patented technology, HX711 is a precision 24-bit analog-to-digital converter (ADC) designed for weigh scales and industrial control applications to interface directly with a bridge sensor.

The input multiplexer selects either Channel A or B differential input to the low-noise programmable gain amplifier (PGA). Channel A can be programmed with a gain of 128 or 64, corresponding to a full-scale differential input voltage of $\pm 20\text{mV}$ or $\pm 40\text{mV}$ respectively, when a 5V supply is connected to AVDD analog power supply pin. Channel B has a fixed gain of 32. On-chip power supply regulator eliminates the need for an external supply regulator to provide analog power for the ADC and the sensor. Clock input is flexible. It can be from an external clock source, a crystal, or the on-chip oscillator that does not require any external component. On-chip power-on-reset circuitry simplifies digital interface initialization.

There is no programming needed for the internal registers. All controls to the HX711 are through the pins.

FEATURES

- Two selectable differential input channels
- On-chip active low noise PGA with selectable gain of 32, 64 and 128
- On-chip power supply regulator for load-cell and ADC analog power supply
- On-chip oscillator requiring no external component with optional external crystal
- On-chip power-on-reset
- Simple digital control and serial interface: pin-driven controls, no programming needed
- Selectable 10SPS or 80SPS output data rate
- Simultaneous 50 and 60Hz supply rejection
- Current consumption including on-chip analog power supply regulator:
 - normal operation $< 1.5\text{mA}$, power down $< 1\mu\text{A}$
- Operation supply voltage range: 2.6 ~ 5.5V
- Operation temperature range: -40 ~ +85°C
- 16 pin SOP-16 package

APPLICATIONS

- Weigh Scales
- Industrial Process Control

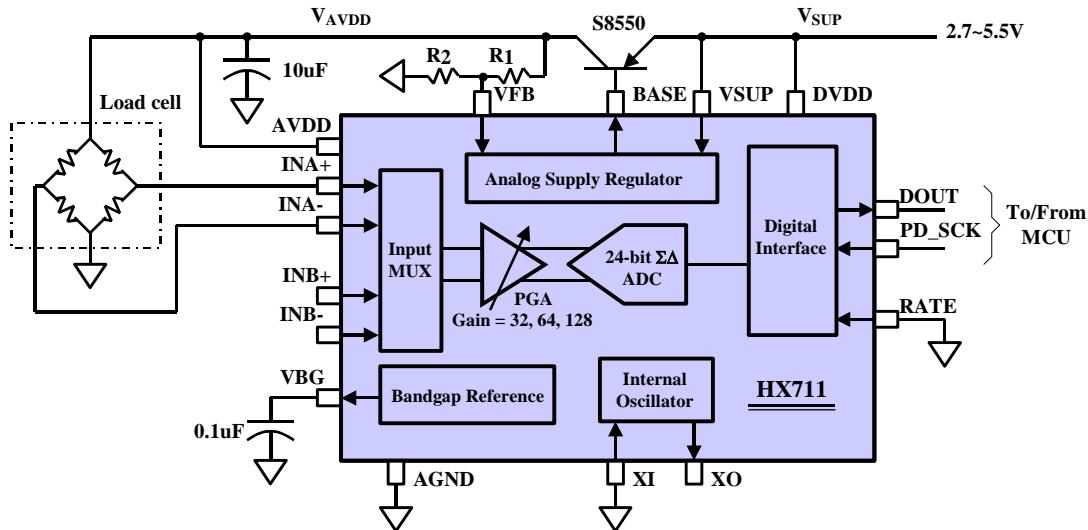


Fig. 1 Typical weigh scale application block diagram

Pin Description

Regulator Power	VSUP		1	•	16	DVDD	Digital Power
Regulator Control Output	BASE		2		15	RATE	Output Data Rate Control Input
Analog Power	AVDD		3		14	XI	Crystal I/O and External Clock Input
Regulator Control Input	VFB		4		13	XO	Crystal I/O
Analog Ground	AGND		5		12	DOUT	Serial Data Output
Reference Bypass	VBG		6		11	PD_SCK	Power Down and Serial Clock Input
Ch. A Negative Input	INNA		7		10	INPB	Ch. B Positive Input
Ch. A Positive Input	INPA		8		9	INNB	Ch. B Negative Input

SOP-16L Package

Pin #	Name	Function	Description
1	VSUP	Power	Regulator supply: 2.7 ~ 5.5V
2	BASE	Analog Output	Regulator control output (NC when not used)
3	AVDD	Power	Analog supply: 2.6 ~ 5.5V
4	VFB	Analog Input	Regulator control input (connect to AGND when not used)
5	AGND	Ground	Analog Ground
6	VBG	Analog Output	Reference bypass output
7	INA-	Analog Input	Channel A negative input
8	INA+	Analog Input	Channel A positive input
9	INB-	Analog Input	Channel B negative input
10	INB+	Analog Input	Channel B positive input
11	PD_SCK	Digital Input	Power down control (high active) and serial clock input
12	DOUT	Digital Output	Serial data output
13	XO	Digital I/O	Crystal I/O (NC when not used)
14	XI	Digital Input	Crystal I/O or external clock input, 0: use on-chip oscillator
15	RATE	Digital Input	Output data rate control, 0: 10Hz; 1: 80Hz
16	DVDD	Power	Digital supply: 2.6 ~ 5.5V

Table 1 Pin Description

KEY ELECTRICAL CHARACTERISTICS

Parameter	Notes	MIN	TYP	MAX	UNIT
Full scale differential input range	V(inp)-V(inn)		$\pm 0.5(\text{AVDD/GAIN})$		V
Common mode input		AGND+1.2		AVDD-1.3	V
Output data rate	Internal Oscillator, RATE = 0		10		Hz
	Internal Oscillator, RATE = DVDD		80		
	Crystal or external clock, RATE = 0		$f_{\text{clk}}/1,105,920$		
	Crystal or external clock, RATE = DVDD		$f_{\text{clk}}/138,240$		
Output data coding	2's complement	800000		7FFFFFF	HEX
Output settling time ⁽¹⁾	RATE = 0		400		ms
	RATE = DVDD		50		
Input offset drift	Gain = 128		0.2		mV
	Gain = 64		0.4		
Input noise	Gain = 128, RATE = 0		50		nV(rms)
	Gain = 128, RATE = DVDD		90		
Temperature drift	Input offset (Gain = 128)		± 6		nV/°C
	Gain (Gain = 128)		± 5		ppm/°C
Input common mode rejection	Gain = 128, RATE = 0		100		dB
Power supply rejection	Gain = 128, RATE = 0		100		dB
Reference bypass (V _{BG})			1.25		V
Crystal or external clock frequency		1	11.0592	20	MHz
Power supply voltage	DVDD	2.6		5.5	V
	AVDD, VSUP	2.6		5.5	
Analog supply current (including regulator)	Normal		1400		μA
	Power down		0.3		
Digital supply current	Normal		100		μA
	Power down		0.2		

(1) Settling time refers to the time from power up, reset, input channel change and gain change to valid stable output data.

Table 2 Key Electrical Characteristics

Analog Inputs

Channel A differential input is designed to interface directly with a bridge sensor's differential output. It can be programmed with a gain of 128 or 64. The large gains are needed to accommodate the small output signal from the sensor. When 5V supply is used at the AVDD pin, these gains correspond to a full-scale differential input voltage of $\pm 20\text{mV}$ or $\pm 40\text{mV}$ respectively.

Channel B differential input has a fixed gain of 32. The full-scale input voltage range is $\pm 80\text{mV}$, when 5V supply is used at the AVDD pin.

Power Supply Options

Digital power supply (DVDD) should be the same power supply as the MCU power supply.

When using internal analog supply regulator, the dropout voltage of the regulator depends on the external transistor used. The output voltage is equal to $V_{AVDD}=V_{BG}*(R1+R2)/ R1$ (Fig. 1). This voltage should be designed with a minimum of 100mV below VSUP voltage.

If the on-chip analog supply regulator is not used, the VSUP pin should be connected to either AVDD or DVDD, depending on which voltage is higher. Pin VFB should be connected to Ground and pin BASE becomes NC. The external 0.1uF bypass capacitor shown on Fig. 1 at the VBG output pin is then not needed.

Clock Source Options

By connecting pin XI to Ground, the on-chip oscillator is activated. The nominal output data rate when using the internal oscillator is 10 (RATE=0) or 80SPS (RATE=1).

If accurate output data rate is needed, crystal or external reference clock can be used. A crystal can be directly connected across XI and XO pins. An external clock can be connected to XI pin, through a 20pF ac coupled capacitor. This external clock is not required to be a square wave. It can come directly from the crystal output pin of the MCU chip, with amplitude as low as 150 mV.

When using a crystal or an external clock, the internal oscillator is automatically powered down.

Output Data Rate and Format

When using the on-chip oscillator, output data rate is typically 10 (RATE=0) or 80SPS (RATE=1).

When using external clock or crystal, output data rate is directly proportional to the clock or crystal frequency. Using 11.0592MHz clock or crystal results in an accurate 10 (RTE=0) or 80SPS (RATE=1) output data rate.

The output 24 bits of data is in 2's complement format. When input differential signal goes out of the 24 bit range, the output data will be saturated at 800000h (MIN) or 7FFFFFFh (MAX), until the input signal comes back to the input range.

Serial Interface

Pin PD_SCK and DOUT are used for data retrieval, input selection, gain selection and power down controls.

When output data is not ready for retrieval, digital output pin DOUT is high. Serial clock input PD_SCK should be low. When DOUT goes to low, it indicates data is ready for retrieval. By applying 25~27 positive clock pulses at the PD_SCK pin, data is shifted out from the DOUT output pin. Each PD_SCK pulse shifts out one bit, starting with the MSB bit first, until all 24 bits are shifted out. The 25th pulse at PD_SCK input will pull DOUT pin back to high (Fig.2).

Input and gain selection is controlled by the number of the input PD_SCK pulses (Table 3). PD_SCK clock pulses should not be less than 25 or more than 27 within one conversion period, to avoid causing serial communication error.

PD_SCK Pulses	Input channel	Gain
25	A	128
26	B	32
27	A	64

Table 3 Input Channel and Gain Selection

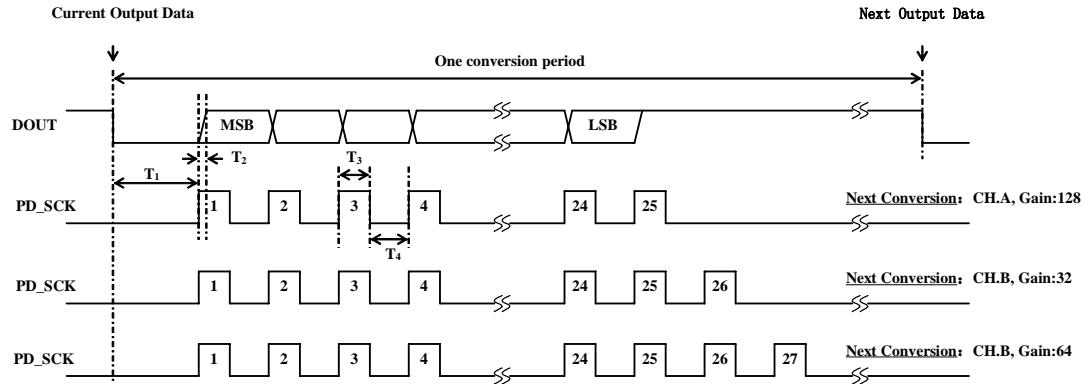


Fig.2 Data output, input and gain selection timing and control

Symbol	Note	MIN	TYP	MAX	Unit
T ₁	DOUT falling edge to PD_SCK rising edge	0.1			μs
T ₂	PD_SCK rising edge to DOUT data ready			0.1	μs
T ₃	PD_SCK high time	0.2	1	50	μs
T ₄	PD_SCK low time	0.2	1		μs

Reset and Power-Down

When chip is powered up, on-chip power on rest circuitry will reset the chip.

Pin PD_SCK input is used to power down the HX711. When PD_SCK Input is low, chip is in normal working mode.

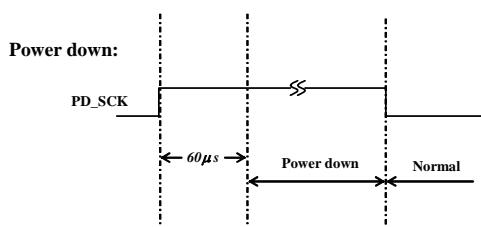


Fig.3 Power down control

When PD_SCK pin changes from low to high and stays at high for longer than 60μs, HX711 enters power down mode (Fig.3). When internal regulator is used for HX711 and the external transducer, both HX711 and the transducer will be

powered down. When PD_SCK returns to low, chip will reset and enter normal operation mode.

After a reset or power-down event, input selection is default to Channel A with a gain of 128.

Application Example

Fig.1 is a typical weigh scale application using HX711. It uses on-chip oscillator (XI=0), 10Hz output data rate (RATE=0). A Single power supply (2.7~5.5V) comes directly from MCU power supply. Channel B can be used for battery level detection. The related circuitry is not shown on Fig. 1.

Reference PCB Board (Single Layer)

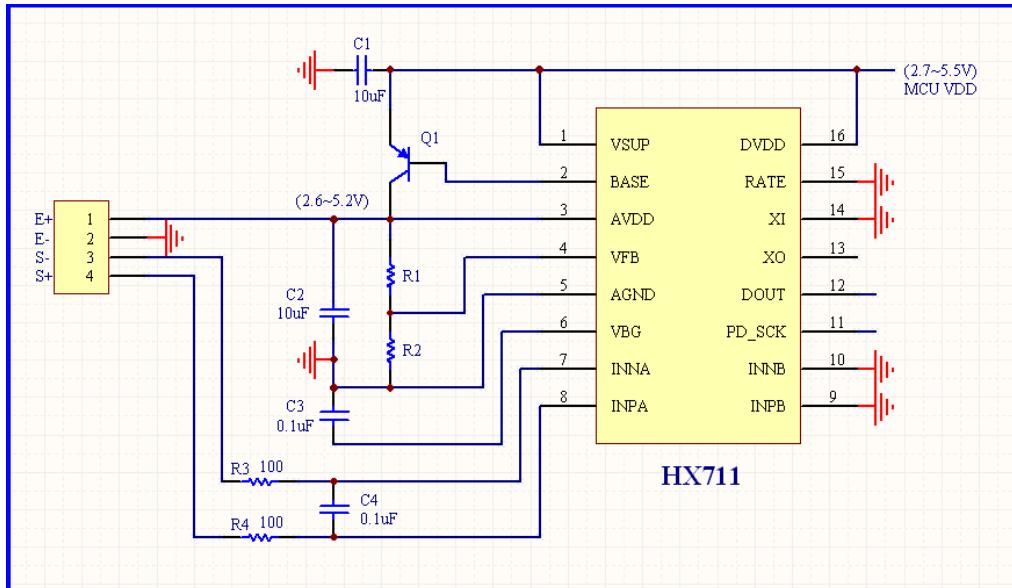


Fig.4 Reference PCB board schematic

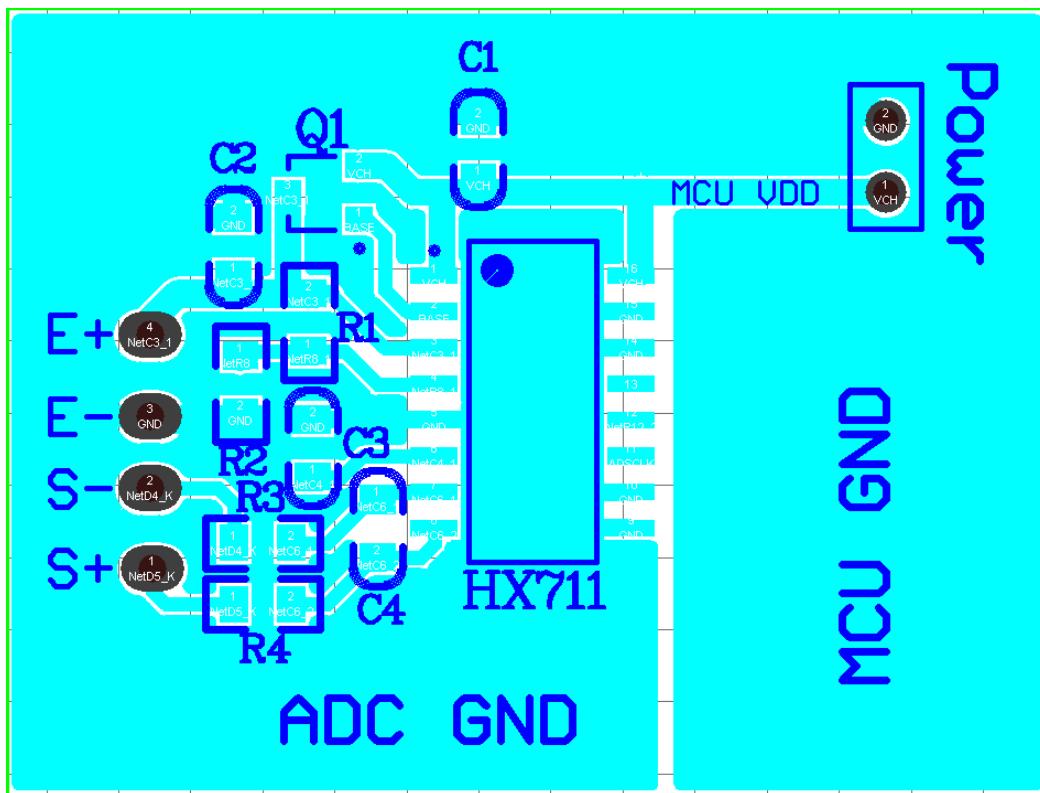


Fig.5 Reference PCB board layout

Reference Driver (Assembly)

```

/*
Call from ASM:      LCALL    ReaAD
Call from C:      extern unsigned long ReadAD(void);

.

.

.

PUBLIC      ReadAD
HX711ROM   segment code
rseg        HX711ROM

sbit        ADDO = P1.5;
sbit        ADSK = P0.0;
/*
OUT:    R4, R5, R6, R7    R7=>LSB
*/
ReadAD:
    CLR    ADSK           //AD Enable (PD_SCK set low)
    SETB   ADDO           //Enable 51CPU I/O
    JB     ADDO, $         //AD conversion completed?
    MOV    R4, #24

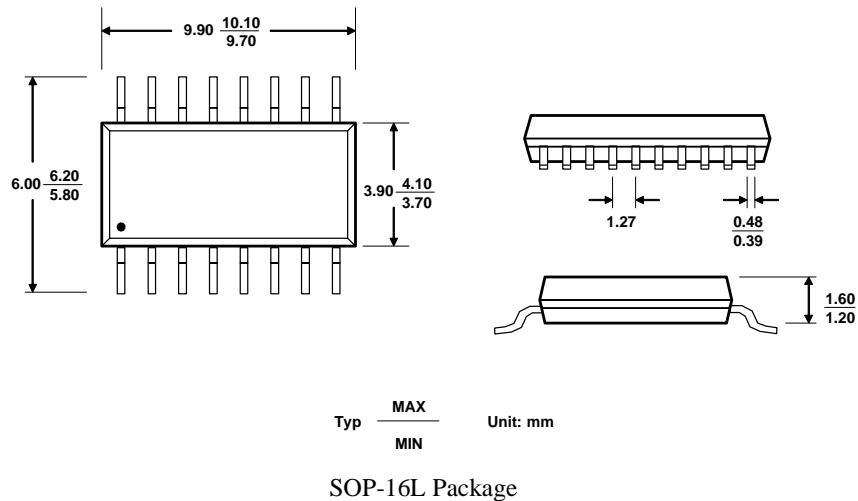
ShiftOut:
    SETB   ADSK           //PD_SCK set high (positive pulse)
    NOP
    CLR    ADSK           //PD_SCK set low
    MOV    C, ADDO         //read on bit
    XCH   A, R7           //move data
    RLC   A
    XCH   A, R7
    XCH   A, R6
    RLC   A
    XCH   A, R6
    XCH   A, R5
    RLC   A
    XCH   A, R5
    DJNZ  R4, ShiftOut    //moved 24BIT?
    SETB   ADSK
    NOP
    CLR    ADSK
    RET
END

```

Reference Driver (C)

```
//-----  
sbit ADD0 = P1^5;  
sbit ADSK = P0^0;  
unsigned long ReadCount(void) {  
    unsigned long Count;  
    unsigned char i;  
    ADD0=1;  
    ADSK=0;  
    Count=0;  
    while(ADD0);  
    for (i=0;i<24;i++) {  
        ADSK=1;  
        Count=Count<<1;  
        ADSK=0;  
        if(ADD0) Count++;  
    }  
    ADSK=1;  
    Count=Count ^ 0x800000;  
    ADSK=0;  
    return(Count);  
}
```

Package Dimensions





This Standard Servo-TowerPro SG-5010(180° Rotation) gives value to your money. This SG-5010 servo is also known for its high specs at a very affordable price to everyone!!!

We have imported this copy of original TowerPro SG-5010 from our trusted supplier. Though it is the copy of the original SG5010 TowerPro servo, this motor shows some of the excellent operational characteristics so we are selling it as **Standard Quality Servo Motor. The good optimized performance and reliability of this servo have made it the favorite choice of many RC hobbyists.**

They are compatible with all the applications that use the genuine TowerPro SG-5010 standard Servo because of their same size, comparable quality, and optimized performance.

It equips Carbon Fiber Gears which makes the servo motor much lighter than same metal gear motor. For small load applications using the metal gear servo motor adds on unnecessary weight, so we suggest using this lightweight plastic gear servo motors. It also equips high-quality bearings which ensure smooth and reliable operations.

The TowerPro SG-5010 Servo rotates 90° in each direction, hence it is a 180° rotation servo. It is a Digital Servo Motor which receives and processes PWM signal faster and better. It equips sophisticated internal circuitry that provides good torque, holding power, and faster updates in response to external forces.

They are packed within a tight sturdy plastic case which makes them water and dust resistant which is a very useful feature in RC planes, Boats, and RC Monster Trucks etc. It equips 3-wire JR servo plug which is compatible with Futaba connectors too.

Wire Description

RED – Positive

Brown – Negative

Orange – Signal

Model	SG5010
Weight(gm)	39
Operating Voltage	4.8 – 6.6 V
Operating Speed @4.8V	0.20sec/60°
Operating Speed @6.6V	0.16sec/60°
Stall Torque @4.8V	5.5 kg-cm

Stall Torque @6.6V	6.5 kg-cm
Operating Temperature (°C)	0 to 55
Dead Band Width	1 μ s
Gear Type	Glass Fiber
Rotational Degree	180
Servo Plug	JR
Cable Length	32 mm
Length (mm)	38
Width (mm)	20
Height (mm)	40
Shipment Weight	0.100 kg
Shipment Dimensions	10 x 8 x 6 cm