Application Note AN2124

A PSoC™ Morse Decoder

By: Melchor A. Varela Morales
Associated Project: Yes
Associated Part Family: CY8C25xxx, CY8C26xxx

Summary

The mixture of PSoC™ analog and digital processing capabilities plus its low cost and ease of use makes it an ideal option for ham projects. This application note describes the development of a Morse decoder based on a PSoC™ device.

Introduction

The PSoC™ family of programmable system-on-chip devices are great for ham's accessory projects because the availability of configurable analog and digital blocks which allow to build complete solutions in only one chip. Besides, the software development tools are free and it is available a low cost emulator.

As an example of the capabilities of PSoC™ for ham projects this application note describes the development of an inexpensive Morse decoder using just the PSoC™ device and an off-the-shelf 2x16 LCD display.

This decoder will automatically decode the Morse symbols and present them in the display using an automatic scroll feature. It features automatic speed tracking and CW speed display. Applications range from the eavesdropping of Morse code QSO's to Morse code learning and practicing.

The decoder has an analog front-end composed of an input band-pass filter and an eight-bit delta-sigma analog to digital converter, all implemented using configurable analog blocks from the PSoC™ architecture. The tone detection and symbol decoding logic is all implemented by software.

Morse Telegraphy

Telegraphy by on-off keying of a carrier is the oldest radio modulation system but it still is in use, especially by hams, because power and bandwidth efficiency, simplicity and its universality. It is also known as CW (for continuous wave).

The Morse code character set defines the characters as a series of 'dit' and 'dash' symbols. The reference symbol duration is the 'dit'. A 'dash' symbol is three 'dits' long. The space between a pair of symbols is one 'dit' long. The space between characters is three 'dits' long, and the space between words is five 'dits' long.

The speed of Morse Telegraphy is usually expressed in WPM, rather than bauds. The following formulas relate WPM to bauds:

- WPM = 2.4 x dots/s
- WPM = 1.2 x B

Where

- WPM = telegraph speed in words per minute
- 2.4 = a constant calculated by comparing dots per second with plain language Morse code sending the word "PARIS"
- B = telegraph speed in bauds

Thus a keying speed of 25 dots/s or 50 baud is equal to 60 WPM.

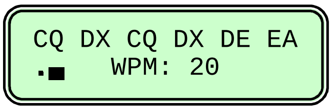
For aural reception, in a typical ham radio receiver, a Morse code RF signal is not completely demodulated to its original dc pulse, because only thumping would be heard. Instead, the signal is moving down to AF. The presented Morse decoder will detect the AF signal, in this case tuned to 800Hz, and it will measure the on-off keying periods that they will interpreted and translated in order to be shown in the display.

Decoder Operation

The decoder requires a 9V or 12V external power supply, and it has to be connected to the receiver speaker output or any other receiver audio output. The design can be easily adapted to use a microphone instead a direct connection to the receiver, and to use batteries instead the power supply.

The operation of the decoder is completely automatic, and no user intervention is required. It is only required to tune the receiver to a Morse signal, and adjust the receiver output level. It is important to adjust the receiver so that the AF Morse signal is centered on 800Hz. The signal coming from a CW beacon is great to start the operation.

The decoder has a 2-lines by 16 columns LCD to present the decoding results. The first line shows the received characters. When the line is completed, on the receiving of the following characters the decoder shifts the display line character by character to the left, so it will always display the last 16 received characters. Following illustration shows the decoder screen:



The second line of the LCD is dedicated to the decoder status. The first column shows the tone detector status. It displays a dot ('.') when a tone is detected and a space when no tone is detected. The second column displays the signal strength level using a vertical bargraph. The remaining columns display the current WPM rate. The decoder restarts to the decoding default values (20WPM) and goes to the idle state if no symbols are detected in a six seconds period. In this idle state the WPM rate is displayed with lower dash symbols.

The decoding capabilities are essentially connected to the received signal quality, and it will not decode weak and vanishing signals with bad signal to noise ratios, but in any case performances are quite satisfactory. It has been tested with speeds from 10 to 40WPM, but this range could be extended with further adjustments.

System Architecture

The decoder design consists of three principle blocks: analog front-end, the tone detector and the symbol detection logic.

The analog front-end digitizes the input audio signal in order to be processed by the tone detector logic. It is composed basically of an input band-pass filter and an eight-bit delta-sigma analog to digital converter, all implemented using PSoC user modules

The tone detector is based in a quadrature correlator implemented by software based on the implementation described in Ref. 1 "Cypress MicroSystems AN2122: DTMF Detector".

The symbol detection logic will decode the Morse code characters by measuring and interpreting the on-off periods. The basis of this process is two moving average filters, one for the 'dit' period and another for the 'dash' period, that provide the automatic speed tracking feature.

Tone Detection

The tone detector is based in a quadrature correlator implemented by software. Correlation is the search for similarities between two sets of data, either a pair of signals (cross-correlation) or self-similarities in the same signal at different times (auto-correlation). Correlation of signals requires the computation of a sum of time-shifted products.

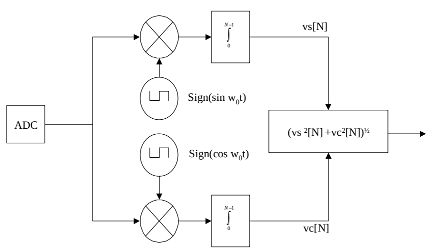


Figure 1, Quadrature Correlator

In this case the correlator correlates the incoming signal with two reference quadrature signals, which correspond to the sign function of sine and cosine signals of the tone to detect. The sign function mimics a square wave of the tone to detect. The advantage of this correlator is the absence of multiplication operations.

The correlator performs addition when the reference signal is positive and subtraction when the reference signal is negative. The quadrature correlator is illustrated in Figure 1.

After processing N samples, the quadrature components are squared to eliminate incoming signal-phase dependence. The following formulas illustrate the correlator operation:

$$vs[n] = \sum_{n=1}^{N} x[n] \cdot sign\left[sin\left(2\pi \frac{fd}{f_s} n \right) \right]$$

$$vc[n] = \sum_{n=1}^{N} x[n] \cdot sign\left[\cos\left(2\pi \frac{fd}{f_s} n \right) \right]$$

$$Z^2 = vs^2[N] + vc^2[N]$$

Where vs[n] and vc[n] are the sine and cosine correlator components, f_d the detection frequency, and f_s the sampling frequency. Sine and cosine accumulators have to be initialized to zero.

The tone detector parameters, assuming the detection frequency f_d to be 800 Hz, have been selected with the following criteria. The N value has been chosen in order to have a detection time of about 5ms, in order to provide enough timing resolution, especially with high speed coding. The sampling frequency has to be at least two times the tone frequency in order to satisfy the Nyquist criteria, and it has been carefully selected in order to have an integer number of samples and signal cycles, taking care too of selecting an integer divisor for the ADC clock.

Choosing a sampling frequency of 6465 sps, and N fixed to 32, we get a detection frequency of 808Hz and a detection time of 4.95ms. In this case the ADC clock is obtained from the 48M clock and a divisor of 29. For N samples we get four signal cycles and eight samples per cycle (45º phased). Thank to this, it is possible to implement very easily the sign function by using the sampling counter, because the sign function changes every four samples. The sine and cosine functions have a 90º phase shift which in this case, it is exactly two counter counts.

Hardware Architecture

The required hardware elements are mostly implemented inside the PSoC device. The external elements are the power supply circuits, the LCD display, and the optional audio amplifier.

Power Supply

It has been decided to power the PSoC at 5 volts in order to have the possibility to operate at its maximum speed, 24 MHz. In any case, this design is operating at 12 MHz, so it could be modified to run at 3.3V. The 5-volt supply is generated from a 7805 linear regulator (U4). The input is protected against inverse polarity because the series diode D1. An external DC wall adapter of 9 V or 12 V is adequate to power the unit.

Audio Amplifier

The audio amplifier (U5) is based on the common LM386. It is powered to 5 volts and the gain is set to 50; that is a comfortable value for this design. The volume control is done by a potentiometer (R5). The output is connected to a small speaker.

LCD Display

The required display is an industry standard LCD display based on Hitachi HD44780, or compatible display driver-bus protocol. The display is connected directly to the PSoC I/O ports, uses the 4-bit interface, so only 7 I/O pins are required. Resistor R1 has to be adapted to the specific display used for optimum contrast, because the required bias setting is not standard for all display types.

PSoC

The PSoC device provides most of the required hardware elements. The decoder requires an analog front-end to digitize the input audio signal in order to be processed by the tone detector logic. It is composed basically of a band-pass filter and an eight-bit delta-sigma analog to digital converter, all implemented using PSoC user modules, see Figure 2. Although in this design is not required, it has been decided to use dynamic configuration in order to prepare the design for further features. As a general rule, and even for designs not requiring dynamic configuration, I recommend to create the designs with a base configuration with the most likely common user modules, and the required loadable configurations.

In this case there is a base configuration with the LCD user module, and the DATA configuration implementing the analog front-end.

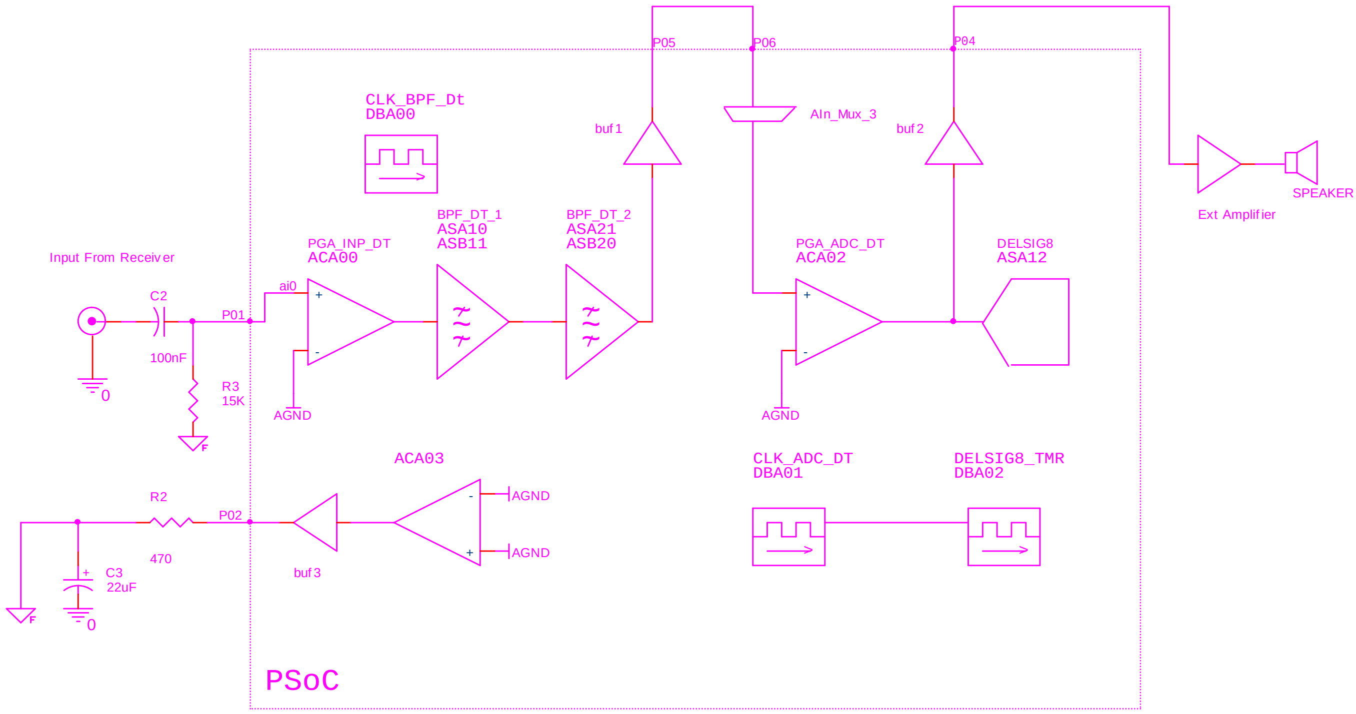


Figure 2, PSoC Configuration

The received AF Morse signal is normally in the 700 to 1000 Hz range. In order to minimize the unwanted frequency components and avoid the aliasing, a four pole bandpass filter is implemented. The filter has a Chebychev response (0.1dB ripple in the pass band) and it has a center frequency of 800Hz and a bandwidth of 800Hz. Filter settings are shown in the following table:

Parameter	Low Pole	High Pole
Q	1,717	1,705
C1	3	5
C2	2	1
C3	4	21
C4	13	4
CA	32	32
CB	32	32
N	138	
Sample Clock	43478 Hz	
Gain	4 dB	

The bandpass filter (BPF_DT_1 and BPF_DT_2) is implemented using four switching capacitor blocks (ASA10 and ASB11 for the low pole, and ASA21 and ASB20 for the high pole). The filter timing reference is set by a Timer 8 user module CLK_BPF_DT (DBA00).

The filter input comes from a PGA user module PGA_INP_DT (ACA00). The gain is set to one, but it can be adapted to the receiver audio output requirements. The input signal is decoupled by an external RC high pass filter (C2 and R3). The proper DC offset is provided by a PGA user module (ACA03) which outputs externally the AGND reference.

The filter output is driven externally to P0[5], and it returns back to the PSoC to the analog input multiplexer 3 via P0[6] in order to be connected to a PGA user module PGA_ADC_DT (ACA02). This PGA provide the interconnection path between the filter and the ADC, and also to the connection to the external amplifier through Buf2 and P0[4].

The analog-to-digital converter is a DELSIG8 user module. This user module uses a switched capacitor block (ASA12) and a digital block (DBA02) to implement a timer. This allows the proper number of integration cycles and the PSoC decimator to process the single bit output stream from the analog block.

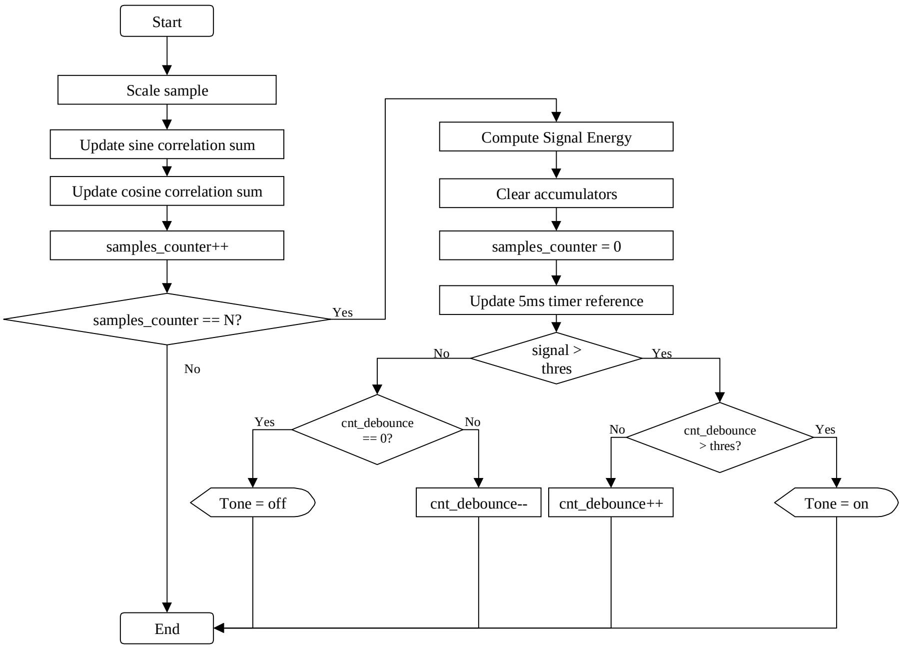


Figure 3, ADC Interrupt Service Routine

The ADC timer requires a prescaler in order to set the required sampling frequency: 6465sps. This timer is implemented using a Timer 8 user module CLK_ADC_DT that is driven by the 48M clock, therefore the divisor is set to 29 according with the sample rate formula for the DELSIG8:

$$SampleRate = \frac{DataClock}{256}$$

Software Architecture

The software is organized around a foreground active level, which performs the symbol detection and presentation tasks. The tone detection logic and timing reference is located in the ADC interrupt service routine. The ADC interrupt service routine is serviced at the sample rate and takes the signal samples to be processed by the tone detection logic. Figure 3 shows the ADC interrupt service routine logic.

The tone detector scales the sample in order to provide a higher dynamic range and multiply it by the sign function of the sine and cosine reference signal and the result is accumulated in the sine and cosine accumulators. The sine and cosine sign reference is obtained from the samples counter by using the third bit of the counter as a sign bit, so every four samples the sign is changed (one reference signal cycle is 8 samples). The 90º phase shift for the cosine reference will be obtained by simply adding two to the sample counter and using the third bit of the counter as for the sine.

After the N samples are processed, the energy is calculated by squaring and summing-up the quadrature components. This is implemented by the integer multiply macros from Application Notes AN2032 and AN2038. The result is divided by 2^{16} for scaling. After, the accumulators are cleared to start a new computation. Because the N value corresponds with a time of 5 ms, it will be used as a time reference for the symbol detection logic.

The signal energy value is used for the signal strength level bargraph, and it will be compared with a fixed value to determinate the presence of a tone. The tone will be validated using a debounce counter. When the signal energy value is above a decision threshold a counter is incremented, until it reach an acceptation value. In that case the tone will be accepted as a valid tone. If the signal is below the decision threshold the counter will be decremented unit it arrives to zero. In that case it will be considered as the end of the tone.

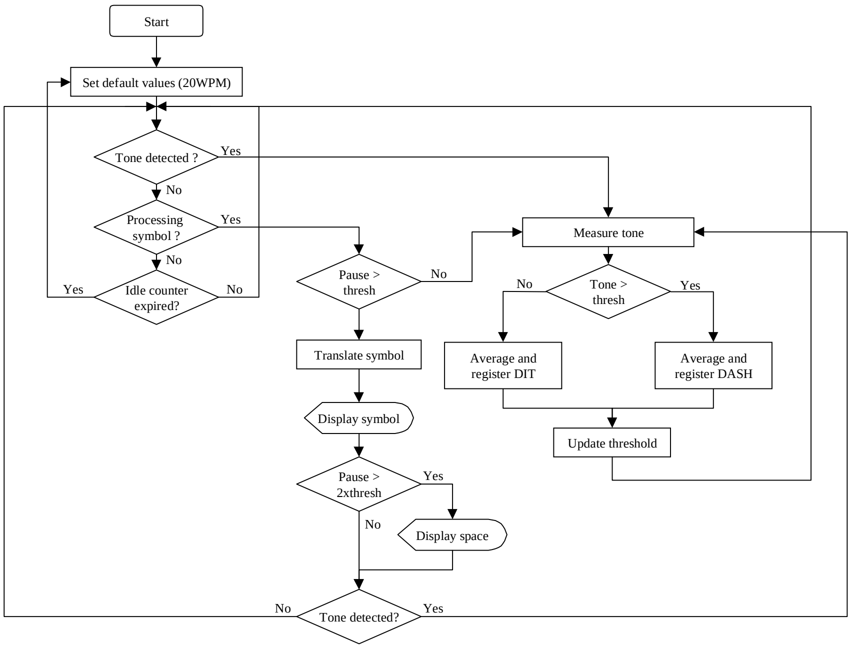


Figure 4, Symbol Decoding Logic

The symbol decoding logic is illustrated in Figure 4. The decoder is continuously looking for the transitions in the tone detection in order to measure the tone and silence duration. The base of the symbol decoding is two running average buffers that store the last four values of the DIT and DASH interval samples. The average values of the DIT and DASH periods are used to set-up the threshold for consecutive symbols. The threshold is the mean value of the DIT and DASH periods. The threshold will be also used for character and word detection by measuring the length of the silences and comparing with this value.

The character decoding is done via a look-up-table indexed by two control variables, which values depend on the previous symbol occurrences, and it is performed when an inter-character or an inter-word silence has been detected. When an inter-word period is detected an automatic space is displayed.

The WPM status is updated in every character occurrence by accessing to a look-up-table indexed by the current average DASH period.

When no symbols are detected in a period of about six seconds, the decoder resets to the default values in order to start the training with another Morse signal. The default decoding values are set for 20WPM.

All of the display control is done by using the LCD user module API. Besides ordinary text functions, the API provides the ability to display bargraphs that are very useful in this case for signal strength level display.

References

1. Cypress MicroSystems AN2122: DTMF Detector
2. Cypress MicroSystems AN2032: Unsigned Multiplication
3. Cypress MicroSystems AN2038: Signed Multi-Byte Multiplication

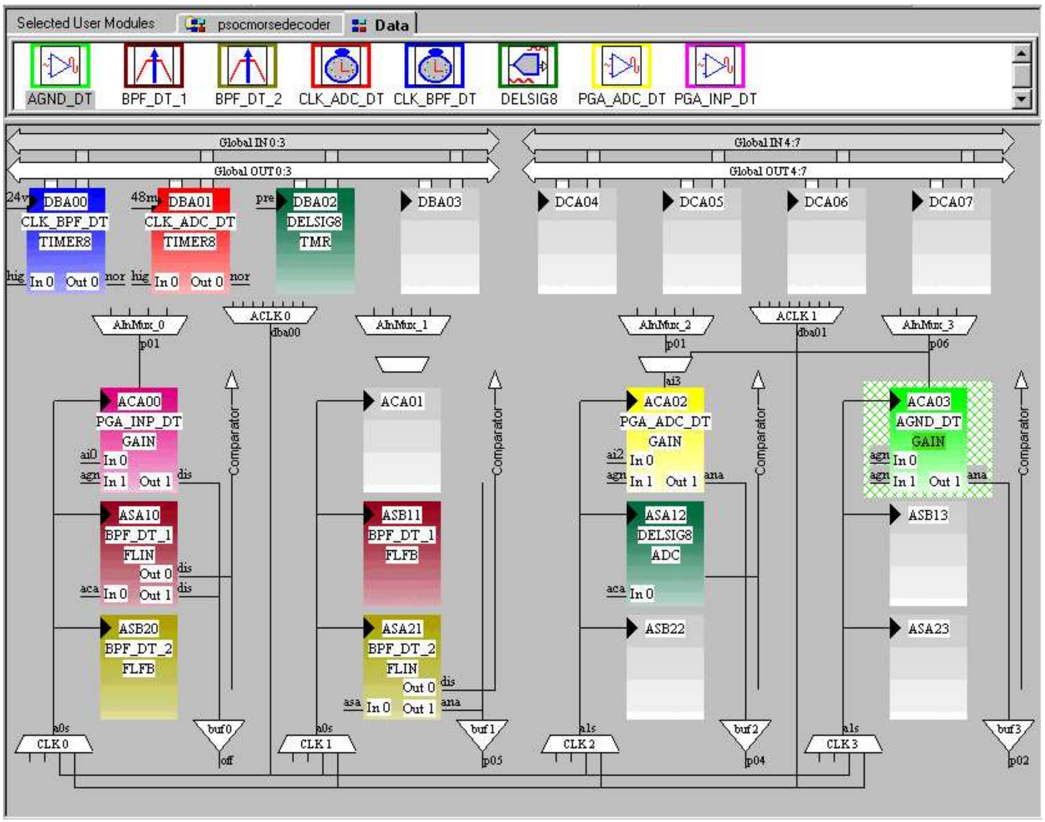


Figure 5, Configuration

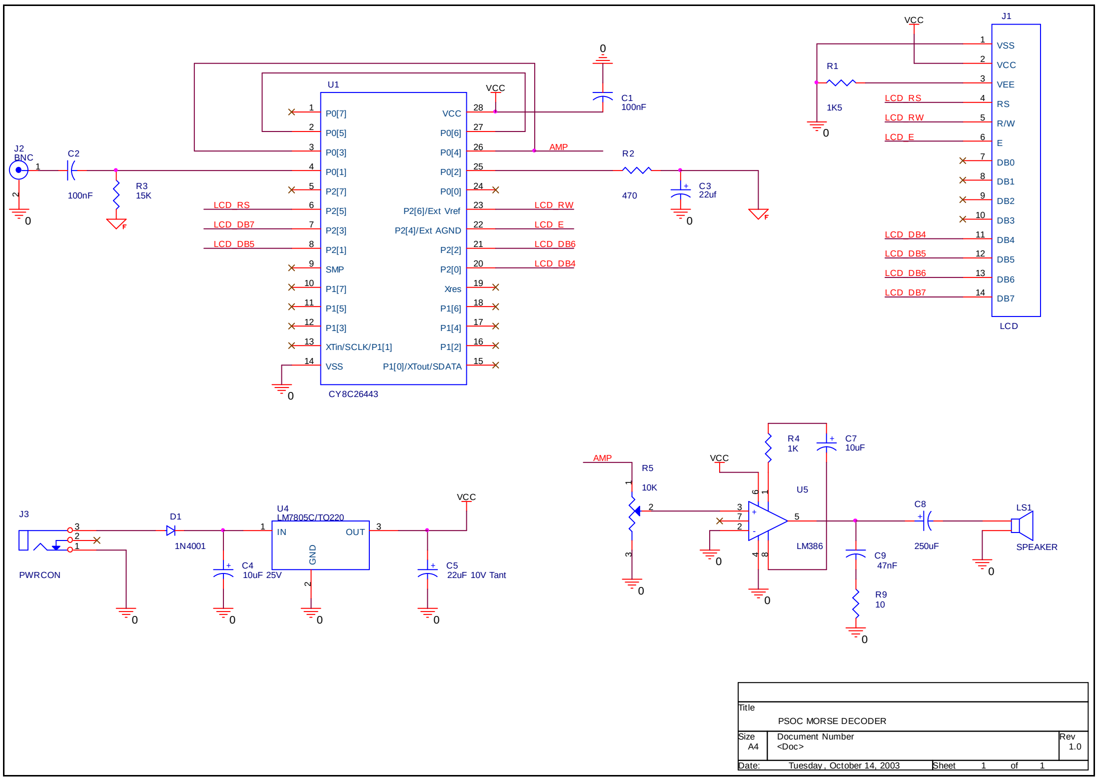


Figure 6, Schematics

Morse Code Character Set

A	● –	Period [.]	● – ● – ● –	Ä	● – ● –
B	– ●●●	Comma [,]	– – ●● – –	Á, Å, À, Â	● – – ● –
C	– ● – ●	Question mark [?]	●● – – ●●	Ç	– ● – ●●
D	– ●●	Hyphen [-]	– ●●●● –	É, È	●● – ●●
E	●	Double dash [=]	– ●●● –	È	● – ●● –
F	●● – ●	Colon [:]	– – – ●●●	Ê	– ●● – ●
G	– – ●	Semicolon [;]	– ● – ● – ●	Ö,Ó	– – – ●
H	●●●●	Left parenthesis [(]	– ● – – ●	Ñ	– – ● – –
I	●●	Right parenthesis [)]	– ● – – ● –	Ü	●● – –
J	● – – –	Fraction bar [/]	– ●● – ●	Ž	– – ●● –
K	– ● –	Quotation marks [“]	● – ●● – ●		
L	● – ●●	Dollar sign [$]	●●● – ●● –		
M	– –	Apostrophe [‘]	● – – – – ●		
N	– ●	Underline [_]	●● – – ● –		
O	– – –	End of message or cross [+]	● – ● – ●		
P	● – – ●	Paragraph [¶]	● – ● – ●●		
Q	– – ● –	End of work [′]	●●● – ● –	0	– – – – –
R	● – ●	Wait [|]	● – ●●●	1	● – – – –
S	●●●	Understood [*]	●●● – ●	2	●● – – –
T	–	Starting signal [➔]	– ● – ● –	3	●●● – –
U	●● –			4	●●●● –
V	●●● –			5	●●●●●
W	● – –			6	– ●●●●
X	– ●● –			7	– – ●●●
Y	– ● – –			8	– – – ●●
Z	– – ●●			9	– – – – ●

About the Author

Name: Melchor A. Varela Morales
Title: Security Expert
Hypercom Corporation
Contact: melchor.varela@gmail.com

Cypress MicroSystems, Inc.
2700 162nd Street SW, Building D
Lynnwood, WA 98037
Phone: 800.669.0557
Fax: 425.787.4641
http://www.cypressmicro.com/ / http://www.cypress.com/aboutus/sales_locations.cfm / support@cypressmicro.com