

Volume

1

A-WIT TECHNOLOGIES INC.

---

... a passion for execution ...

# CS493000 IR Remote Kit Reference Guide Manual

---

Version 1.2

A-WIT TECHNOLOGIES INC.

# CS493000 IR Remote Kit for the C Stamp Reference Manual

---

© A-WIT Technologies Inc.  
Phone (800) 985-AWIT • Fax (800) 985-2948

---

# Table of Contents

Registering Your C Stamp or C Stamp	
Related Product	1
Introduction to the CS493000 IR Remote	
Kit for the C Stamp	1
Detailed Pin Description	3
Principles of IR Remote Communications	4
Notices	8
Getting Support	9
Installing the Microchip MPLAB and C	
Compiler Software	9
Installing the A-WIT C Stamp Quick	
Programmer	9
Installing the USB Software	11
Setting Up the C Stamp Software	
Templates	11
Documentation	11
Creating a First IR Remote C Stamp	
Program	12
Creating the <code>get_remote_key</code> Function	15
Downloading and Running Your Program	17
Creating an IR Remote C Stamp Program	
with Sub-Menus	18
Terms & Conditions	20

---

## Introduction to the CS493000 IR Remote Kit for the C Stamp

**T**he CS493000 IR Remote Kit for the C Stamp allows you to add infrared communication and control capabilities to your C Stamp™ project. The kit contains all the hardware and software necessary for this purpose. Using commands built into the C Stamp software support infrastructure, users can communicate with and control the C Stamp. Using the function recursion capabilities of the WC language, users can also produce sophisticated menus hierarchies with sophisticated controlling behavior.

### Registering Your C Stamp or C Stamp Related Product

At A-WIT Technologies we respect your privacy; however, we do ask you to register your C Stamp or C Stamp related product, so you can receive free of charge product updates. The registration procedure is simple. Just send an e-mail to [tech\\_support@a-wit.com](mailto:tech_support@a-wit.com) with the word “REGISTRATION x” in the subject line, where “x” is the product number that you purchased. If you purchased more than one product, send an e-mail for each different product.

### Introduction to the CS493000 IR Remote Kit for the C Stamp

The CS493000 IR Remote Kit for the C Stamp allows you to add infrared communication and control capabilities to your C Stamp™ project. The kit contains all the hardware and software necessary for this purpose. Using commands built into the C Stamp software support infrastructure, users can communicate with and control the C Stamp. Using the function recursion capabilities of the WC language, users can also produce sophisticated menus hierarchies with sophisticated controlling behavior. The

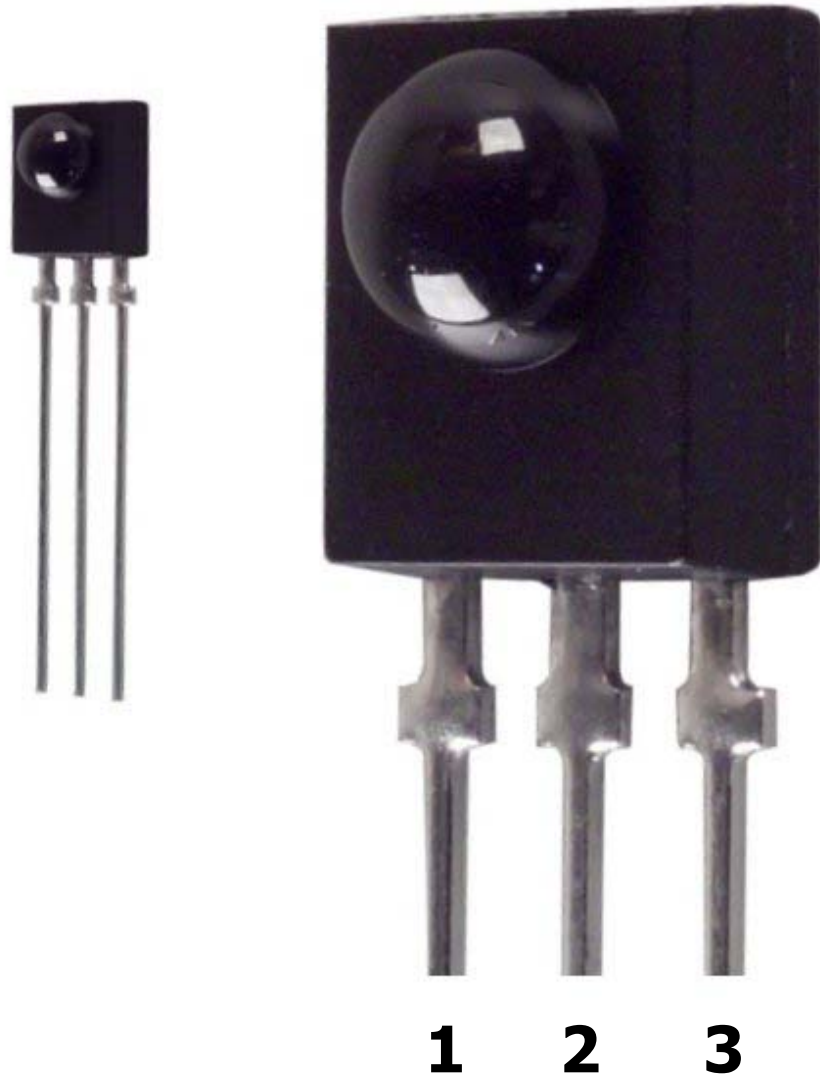
following figure shows the Remote Kit, and the following table details the kit's contents.



<i>IR Remote Kit for the C Stamp Detailed Contents Description</i>	
<i>Item</i>	<i>Minimum Quantity</i>
Remote with Glow-in-the-Dark Buttons	1
Infrared Receiver	1
AAA Batteries	2
17 cm hook-up wire	3
Software and documentation CD	1

## Detailed Pin Description

The figure below shows the pins arrangement of the Infrared Receiver that is included in the Remote Kit, and the following table shows the Detailed Pin Description.

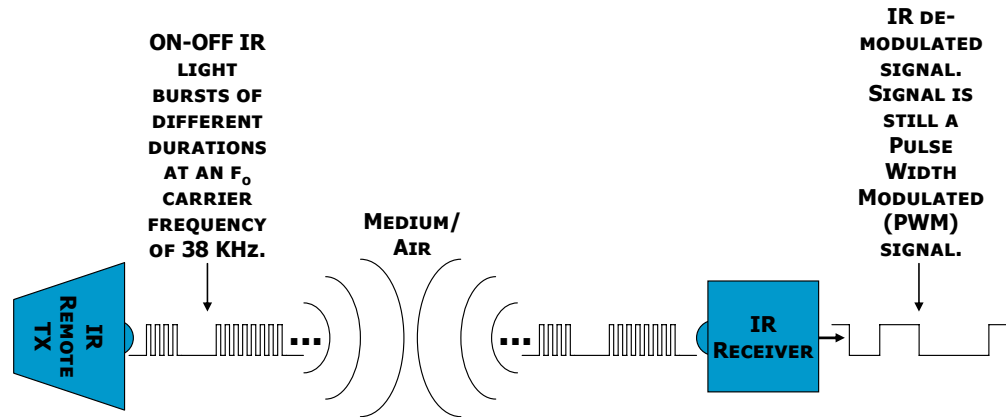


<i>IR Receiver Pin-Out Connectivity</i>		
<i>Pin</i>	<i>Symbol</i>	<i>Description</i>

<i>IR Receiver Pin-Out Connectivity</i>		
<i>Pin</i>	<i>Symbol</i>	<i>Description</i>
1	$V_O$	Output pin.  Connect to any C Stamp I/O pin for digital input into the C Stamp.  A LOW at this pin signifies the detection of IR light with a carrier frequency $f_o$ of 38 KHz. A HIGH at this pin signifies that no IR light is being detected.
2	GND	GND Ground pin.
3	$V_{CC}$	$V_{CC}$ Supply 4.7 – 5.3 V.

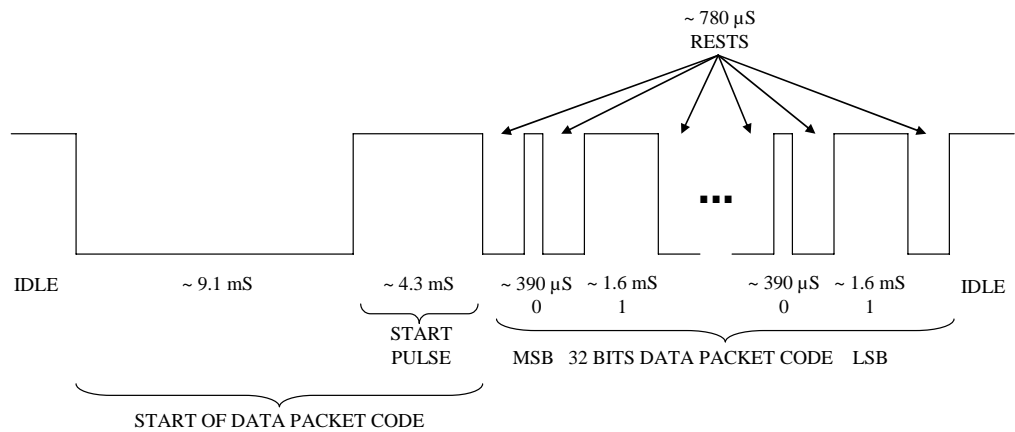
## Principles of IR Remote Communications

The figure below shows the principles of IR Remote communications between the remote and the IR receiver, which is connected to a C Stamp input pin.



Using a properly biased IR LED, the remote sends ON-OF IR light bursts of different durations at a carrier frequency  $f_o$  of 38 KHz. This light travels through the air toward the receiver, which demodulates the IR signaling. The presence of an IR burst is demodulated by the receiver as a digital 0, and the absence of IR light as a digital one. In this manner, the remote encodes identification of a pressed key using Pulse Width Modulation (PWM), where a decoded LOW voltage (or digital 0) by the receiver is

called a REST, a short HIGH (or digital 1) pulse encodes a digital 0, and a long HIGH pulse encodes a digital 1. It is up to the C Stamp to perform the Pulse Width demodulation, as we will show up ahead. The IR receiver only performs the IR demodulation from the original 38 HKz IR carrier. Furthermore, since a LOW out of the receiver conveys the presence of an IR burst, and a HIGH the absence of it, the PWM information is encoded by the remote by the absence or IR light bursts. The PWM protocol out of the IR receiver is shown in the following figure. The timing shown is specific to the remote in the kit, which is a TOSHIBA Model CT-878. The table following the figure shows the specific codes for this remote. Note that the most significant 16 bits of the codes are always the same, and are used for remote identification. Therefore, when working with this remote, it is only sufficient to identify the buttons pressed by the least significant 16 bits, assuming that there are no other remotes pointing at your project. Our interpretation of the narrow pulses for a digital 0 and the wider pulses for a digital 1 is arbitrary. We could have interpreted the narrow pulses for a digital 1 and the wider ones for digital 0's without loss of generality. The codes for the remote would have been simple inverted or 1-complemented.



<b>TOSHIBA CT-878 REMOTE CONTROL CODES</b>		
<b><i>Function/Button/Key</i></b>	<b><i>16 MSBs (in Hex)</i></b>	<b><i>16 LSBs (in Hex)</i></b>
POWER	0x02FD	0x48B7
1	0x02FD	0x807F
2	0x02FD	0x40BF
3	0x02FD	0xC03F

<b>TOSHIBA CT-878 REMOTE CONTROL CODES</b>		
<i>Function/Button/Key</i>	<i>16 MSBs (in Hex)</i>	<i>16 LSBs (in Hex)</i>
4	0x02FD	0x20DF
5	0x02FD	0xA05F
6	0x02FD	0x609F
7	0x02FD	0xE01F
8	0x02FD	0x10EF
9	0x02FD	0x906F
0	0x02FD	0x00FF
100	0x02FD	0x50AF
RECALL	0x02FD	0x38C7
SLEEP	0x02FD	0xA857
CH+	0x02FD	0xD827
MUTE	0x02FD	0x08F7
VOL-	0x02FD	0x7887
VOL+	0x02FD	0x58A7
CH-	0x02FD	0xF807
MENU/ENTER	0x02FD	0x01FE
CAP/TEXT	0x02FD	0xEA15
1/2	0x02FD	0xC639
EXIT	0x02FD	0x1AE5
RESET	0x02FD	0x30CF
CH RTN	0x02FD	0xE817

<b><i>TOSHIBA CT-878 REMOTE CONTROL CODES</i></b>		
<b><i>Function/Button/Key</i></b>	<b><i>16 MSBs (in Hex)</i></b>	<b><i>16 LSBs (in Hex)</i></b>
MTS	0x02FD	0xC837
TV/VIDEO	0x02FD	0xF00F
FAV v	0x02FD	0xB847
FAV ^	0x02FD	0x9867
16:9	0x02FD	0x2AD5

## Getting Started

This chapter is a quick start guide to using the CS493000 IR Remote Kit for the C Stamp. This assumes you have a C Stamp and an appropriate connection kit or development board with the RESET and START circuitry, and the IR receiver output properly connected to the C Stamp. You will also need a programming cable, power supply, PC running Windows® 2000/XP/Media, with a quantity of RAM recommended for the OS, sufficient free hard disk drive space for the software installations, CD-ROM drive, Internet access (recommended only), and available port compatible with your programming cable.

### Notices

CSTAMP™ and CSTAMP™ Related Hardware Products, Software Products and Documentation are developed and distributed by A-WIT Technologies, Inc. All rights reserved by A-WIT Technologies, Inc. A-WIT SOFTWARE OR FIRMWARE AND LITERATURE IS PROVIDED “AS IS,” WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL A-WIT BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY ARISING OUT OF OR IN CONNECTION WITH THE SOFTWARE OR FIRMWARE OR THE USE OF OTHER DEALINGS IN THE SOFTWARE OR FIRMWARE.

MPLAB C-18 and MPLAB C-18 Users Guide is reproduced and distributed by A-WIT Technologies, Inc. under license from Microchip Technology Inc. All rights reserved by Microchip Technology Inc. MICROCHIP SOFTWARE OR FIRMWARE AND LITERATURE IS PROVIDED “AS IS,” WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED, INCLUDING BUT NOT LIMITED TO THE WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE AND NONINFRINGEMENT. IN NO EVENT SHALL MICROCHIP BE LIABLE FOR ANY CLAIM, DAMAGES OR OTHER LIABILITY ARISING OUT OF OR IN

CONNECTION WITH THE SOFTWARE OR FIRMWARE OR THE USE OF OTHER DEALINGS IN THE SOFTWARE OR FIRMWARE.

## Getting Support

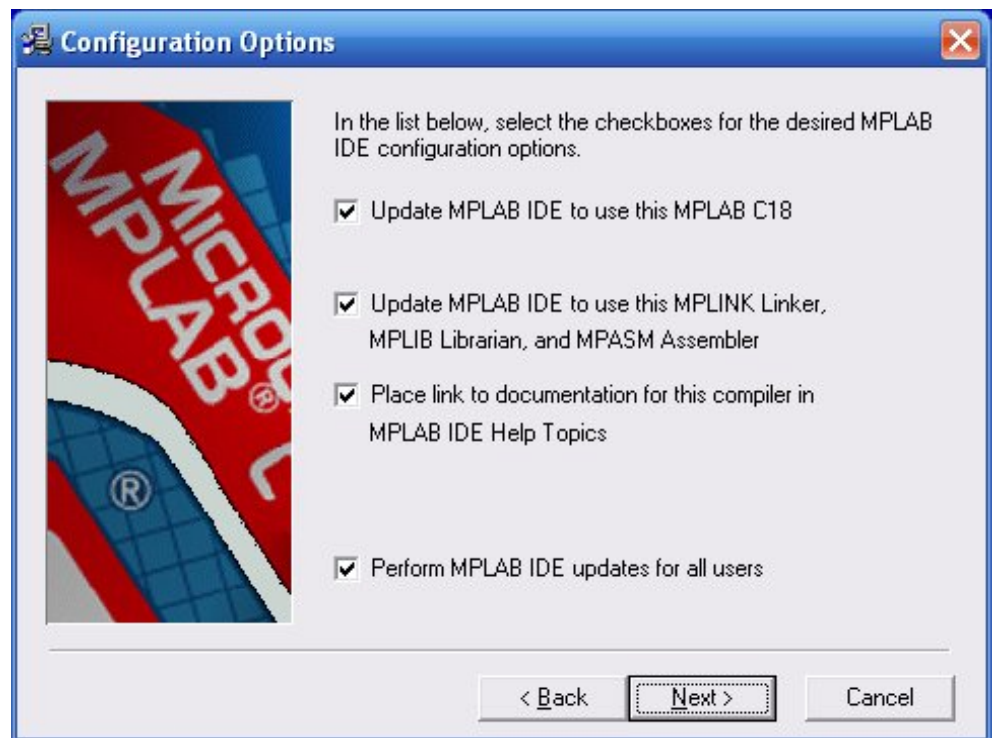
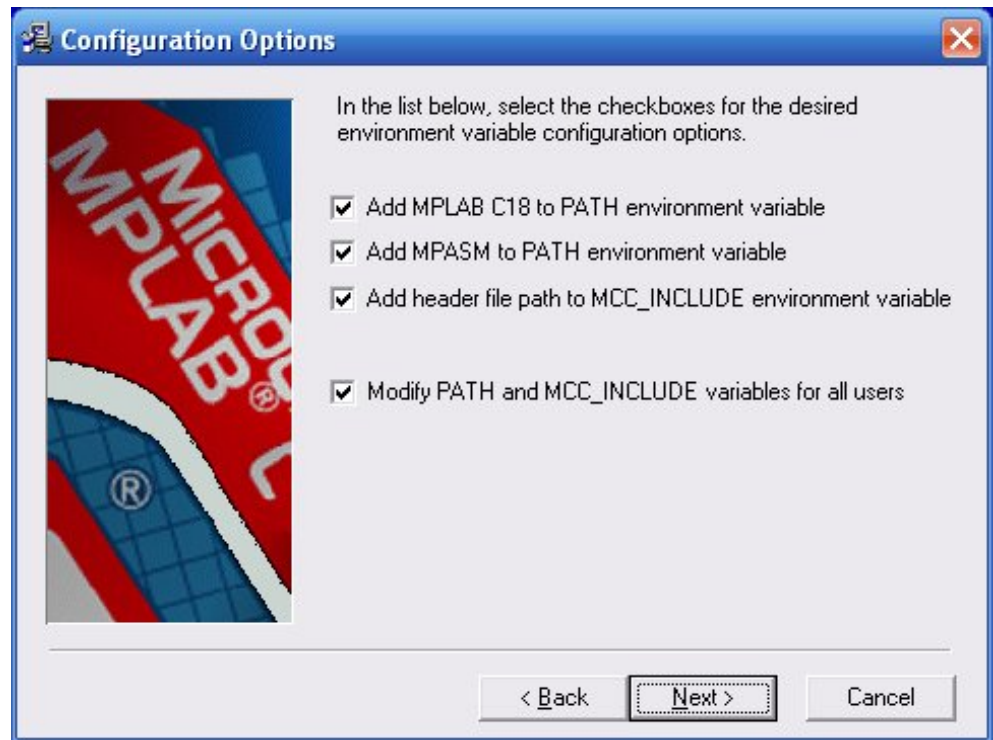
If possible, please check the C Stamp website [www.c-stamp.com](http://www.c-stamp.com) under SUPPORT for any updates to documentation, changes, or notices that may have become available since your Installation CD was produced. If you continue to have any issues for which a solution is not found in the aforementioned website, please e-mail [tech\\_support@a-wit.com](mailto:tech_support@a-wit.com) for help.

## Installing the Microchip MPLAB and C Compiler Software

The first step is to install the Microchip MPLAB software that you will use to develop your programs.

Insert your A-WIT provided Installation CD in your CD drive. Go to the MPLAB directory in the CD and double click on the “MPLAB vX.XX Install” file in that directory. Follow the installation steps, prompts, and directions provided by the installer software, accepting all the default options.

After the MPLAB installation is complete, switch to the C18 directory in the CD, and double click on the file in that directory. Follow the installation steps, prompts, and directions provided by the installer software, accepting all the default options. The only exceptions to accepting all the default options is that on the 5<sup>th</sup> and 6<sup>th</sup> windows of the installation process for the C18 Compiler, you have to select everything as shown in the figures below. This will ensure that MPLAB is configured to use the C18 Compiler.



## Installing the A-WIT C Stamp Quick Programmer

To install the A-WIT C Stamp Quick Programmer, switch to the CSTAMPQP directory in the CD using Windows Explorer, and double click on the file in that directory. Follow the installation steps, prompts, and directions provided by the installer software, accepting all the default options.

## Installing the USB Software

If you purchased a product with a USB download cable, make sure that the A-WIT provided CD is in the CD drive of your PC and insert the USB cable in the USB port of your PC. Windows auto detects the new USB device. If Windows prompts you to install drivers for the USB cable device, follow the installation steps, prompts, and directions provided by the installer software, accepting all the default options.

After the USB adapter has been installed, open a Windows Explorer window from the Accessories sub-menu in the Start menu, and right click on My Computer. Proceed to select Properties, and then select the Hardware tab. Click on the Device Manger button, and expand the Ports (COM & LPT) branch. Make a note of the COM port that has been assigned to the USB-to-Serial adapter. This is the port that should be selected in the C Stamp programmer software.

## Setting Up the C Stamp Software Templates

To set up the C Stamp Software Templates, switch to the CSTAMP\_Template directory in the CD using Windows Explorer, and double click on the file in that directory. Follow the installation steps, prompts, and directions provided by the installer software, accepting all the default options.

## Documentation

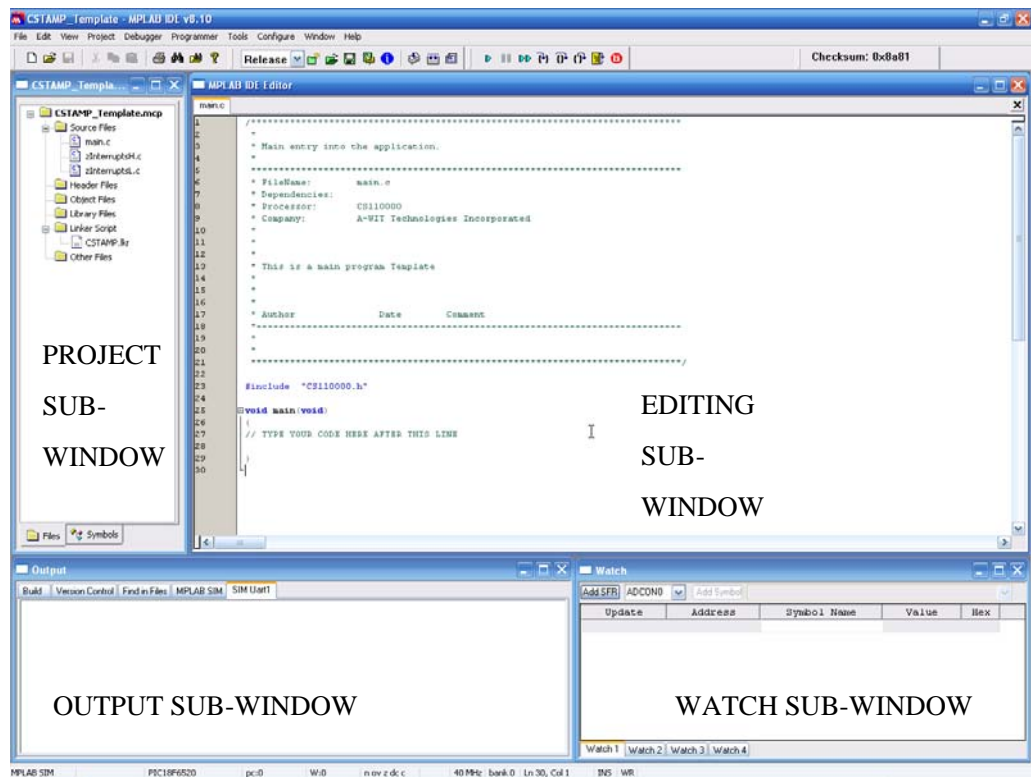
Copy the DOCS directory from the C Stamp Installation CD to your C:\A-WIT directory. This directory contains all the C Stamp related documentation in PDF format.

## Creating a First IR Remote C Stamp Program

This section describes how to create a program with a simple menu for the IR Remote using the functionality of the **PULSEIN** command. Create a directory where you want to have all the files for your program; for example REMOTE1\_APP. We recommend making this directory under your C:\A-WIT directory, so you can have all your CSTAMP related files in one place.

Copy the all files in your C Stamp Software Templates directory C:\A-WIT\CSTAMP\_Template to the directory you just made.

Open the Microchip MPLAB IDE application. As shown the following figure, the IDE has several sub-windows. Depending on the resolution of your screen, your sub-windows may have a different layout. However; you can move and resize these into the position that you want to fit your screen, and your layout for that particular project will get saved upon answering yes to the prompt of saving the workspace when you exit the software development environment.



Go to the “File” menu to “Open Workspace...”. Then navigate to your program directory, and open CSTAMP\_Template.mcw.

Right click on CSTAMP\_Template.mcp in the “Project” sub-window, and “Save as...” the name of your program project after you have navigated to your program directory. For example, your program project could be named “REMOTE1\_APP”. Now when you open the Microchip MPLAB IDE (Integrated Development Environment), and go to your program directory to open the workspace for your program, you will see a .mws file with the name of your program preceding it. This is the file that you should open any time you want to work on your program.

Double click on the main.c source and type the following code fragment where it is indicated. You can omit the comments for brevity, as they are written here to offer clarifications of what the code does. Do pay attention; however, to the indentation of the code blocks between curly brackets for loops, if statements, etc. Although indenting the code is not a requirement for the compiler to parse your code (i.e. any blank spaces are ignored by the compiler), it does help tremendously to make your code much more readable, and consequently, it makes finding any errors easier. Keywords and C Stamp function names in the code fragment below are bolded.

After you START the C Stamp in user mode as explained in the “Downloading and Running Your Program” section (this will not be the RESET/BOOT/DOWNLOAD mode), the program will run. This program assumes that you are using the remote to control a robot to perform the following set of movements with the keys noted in the table below. It also assumes that you have functions to perform these movements. The IR receiver should also be properly connected. Specifically, the program assumes that the output of the IR receiver is connected to Pin 3 of the C Stamp.

<i>Assumed Remotely Controlled Robot Movements</i>			
<i>Function/Button/Key</i>	<i>16 LSBs (Hex)</i>	<i>Movement</i>	<i>Function Called</i>
CH+	0xD827	Go Forward	goforth
CH-	0xF807	Go Back	goback
VOL-	0x7887	Left Turn	turnl
VOL+	0x58A7	Right Turn	turnr
SLEEP	0xA857	Stop	none

The program will first get the key code from the remote, and then set an action given the key code. Then it will perform some movement of the remote according to the

action previously set. The usage of two switch construct allows the program to continuously move the robot according to the last remote key pressed, while also interrogating the remote periodically for a new key code command. The program executes this sequence indefinitely until you reset it by pushing the RESET button.

```

WORD key_code;
BYTE action;
BYTE IRRxPin = 3;
WORD step = 8;
BYTE lp = 18; // left servo pin
BYTE rp = 19; // right servo pin

while(1){
// Get key
    key_code = get_remote_key(IRRXPin);

// Set action according to key
    switch(key_code){
        case 0xD827:
            action = 'F';
            break;
        case 0xF807:
            action = 'B';
            break;
        case 0x7887:
            action = 'L';
            break;
        case 0x58A7:
            action = 'R';
            break;
        case 0xA857:
            action = 'S';
    }
// Move robot
    switch(action){
        case 'F':
            goforth(lp, rp, step); // go forward some
            break;
        case 'B':
            goback(lp, rp, step); // go back some
            break;
        case 'L':
            turnl(lp, rp, 90); // turn left 90 degrees
            PAUSE(250);
            break;
    }
}

```

```

case 'R':
    turnr(lp, rp, 90); // turn right 90 degrees
    PAUSE(250);
}

```

Save your program from the “File” menu or by clicking on the appropriate icon in the tool bar. Then “Build All” from the “Project” menu or from the tool bar.

If the code was typed correctly, you will have a file in your program directory with the name of your program project and a .HEX extension. An example is REMOTE1\_APP.HEX. This is the file that you will download to the C Stamp, as explained up ahead.

If you get an error message or an indication that your program did not build successfully in the “Output” sub-window of the IDE, you probably have one or more syntax error(s). Double click on the line of the “Output” sub-window that mentions the error, and the program line that most likely contains the error will be indicated in the sub-window where you were editing your program. Correct as necessary, and “Build All” again until you get a successful .HEX file output.

## Creating the `get_remote_key` Function

This section describes how to create a function to get a key code from the remote, like the function used in the previous program. This function uses the PULSEIN command of the C Stamp, and is shown below. You can omit the comments for brevity, as they are written here to offer clarifications of what the code does. Do pay attention; however, to the indentation of the code blocks between curly brackets for loops, if statements, etc. Although indenting the code is not a requirement for the compiler to parse your code (i.e. any blank spaces are ignored by the compiler), it does help tremendously to make your code much more readable, and consequently, it makes finding any errors easier. Keywords and C Stamp function names in the code fragment below are bolded.

```
get_remote_key
```

```
WORD get_remote_key(BYTE pin);
```

The `get_remote_key` function returns the PWM demodulated key code based on the IR demodulated signal coming into the C Stamp on a Capture pin. If a key press is not detected or an error occurs, the function returns `FALSE`.

`pin` is a variable/constant/expression that specifies the I/O pin to use. This pin will be set to input mode.

If an error occurs or no key press is detected, the function returns **FALSE**; otherwise the key code is returned. Remember that this code is the 16 Least Significant Bits of the 32-bit code sent by the remote.

FUNCTION:

```

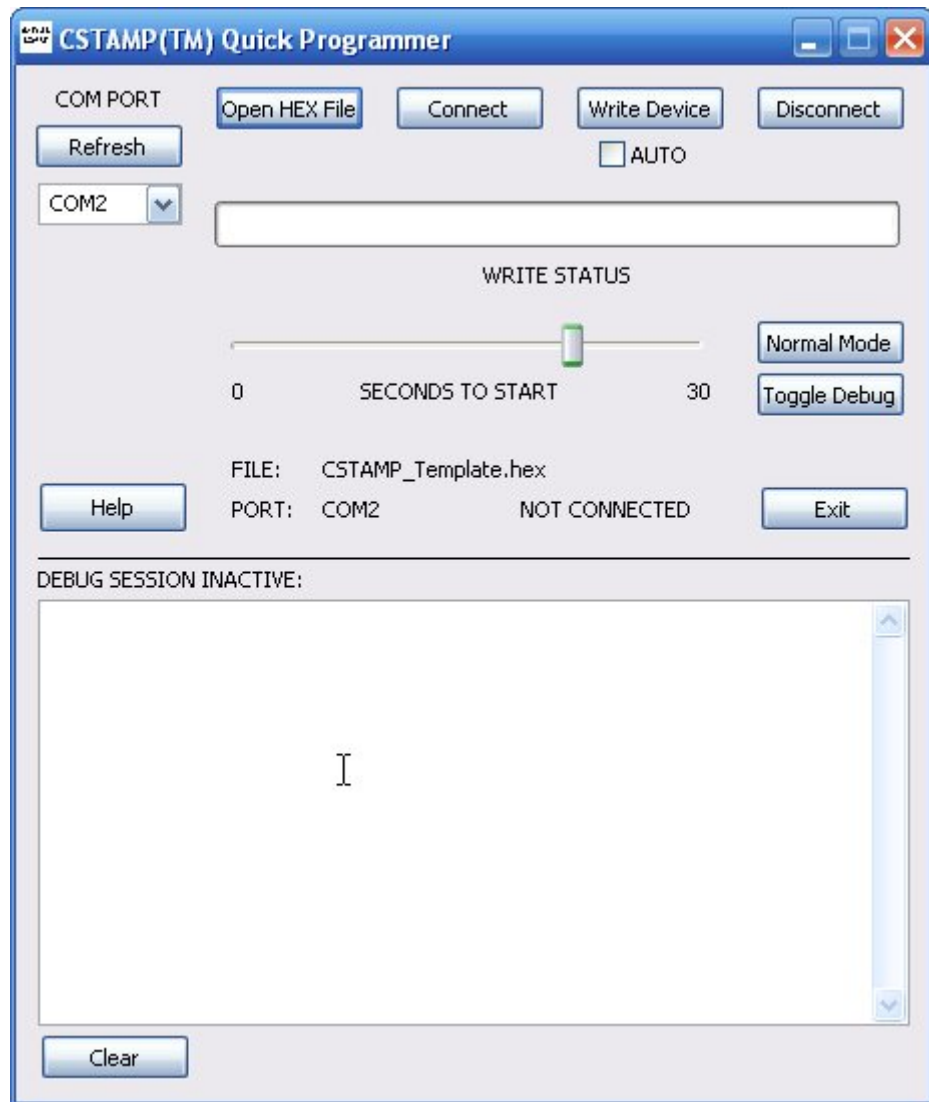
WORD get_remote_key(BYTE pin)
{
    WORD code = 0;
    BYTE i;
    WORD PW;

    if(GTPIND(pin)) return code; // remote not active
    while(TRUE){ // wait for START pulse
        if(PULSIN(pin, 1, 19) > 3750) break;
    }
    // get 16 MSBs and discard
    for(i=0; i<16; i=i+1) PULSIN(pin, 1, 2);
    // get 16 LSBs
    for(i=0; i<16; i=i+1){
        PW = PULSIN(pin, 1, 2);
        code = code << 1;
        if(PW > 1250) code = code | 0x0001;
    }
    return code;
}

```

## Downloading and Running Your Program

Power up your KIT, and connect the KIT to the PC with the provided cable. Upon power up, the C Stamp will be in RESET/BOOT/DOWNLOAD mode. To go back to this mode at any time, just push and let go of the RESET button. Then open the A-WIT C Stamp Quick Programmer application shown in the next figure.



The first step is to choose the serial port that you are using from the drop-down menu. Then click on “Refresh”, so that the program registers your selection. Your selection should show in the status area of the program next to “PORT:”. Then click on “Open HEX File” and load/select the HEX file that you had previously created during the

development of your program. The status area should indicate that the file has been loaded successfully. This is what will be downloaded to the C Stamp. Then click on “Connect”, and the PC will be connected to the KIT, and the status area should indicate so. To download the HEX file to the C Stamp, just click on “Write Device”, and you should see the progress bar after a few seconds, as the HEX file is downloaded. At this point, you can click on “Disconnect” to disconnect the PC from the KIT, disconnect the serial cable from both the PC and the KIT, and start your program manually at the KIT. To do this just push and let go of the RESET button while pushing the START button. Then you can let go of the START button. Alternatively, you can click on “Normal Mode” to start your program from the PC. This will also disconnect the program/PC from the KIT. Then you can disconnect the serial cable from the PC and the KIT. You can also instruct the CSTAMP™ Quick Programmer to wait several seconds before starting your program from the PC and disconnecting by adjusting the “SECONDS TO START” slide. This feature is useful in case you want to keep the PC connected with the serial cable, but need time to manually set up something in a circuit that you have built. If this is not the case it can just be left at the default of “0”, and your program will start from the PC right away. After you click “Normal Mode” and your program is started, the CSTAMP™ Quick Programmer will not be communicating with the C Stamp any longer, so if you want to reconnect, you must click on “Connect” again.

## Creating an IR Remote C Stamp Program with Sub-Menus

The first remote program presented above can be easily modified to include any number of levels of nested submenus. This takes advantage of the powerful WC switch-case construct and the capability of the C Stamp to have the user nest constructs down to any number of levels.

A general skeleton for nested menus is presented below. You can omit the comments for brevity, as they are written here to offer clarifications of what the code does. Do pay attention; however, to the indentation of the code blocks between curly brackets for loops, if statements, etc. Although indenting the code is not a requirement for the compiler to parse your code (i.e. any blank spaces are ignored by the compiler), it does help tremendously to make your code much more readable, and consequently, it makes finding any errors easier. Keywords and C Stamp function names in the code fragment below are bolded.

```

WORD key_code;
BYTE action;
BYTE IRRxPin = 3;
// other variables

while(1){

```

```
// Get key
    key_code = get_remote_key(IRRXPin);

// Set action according to key
    switch(key_code){
        case 0xD827: // or other code
            action = 1;
            break;
        case 0xF807: // or other code
// nested menu or sub-menu
// Get key
            key_code = get_remote_key(IRRXPin);
            switch(key_code){
                case 0xD827: // or other code
                    action = 2;
                    break;
// other cases or further nesting of sub-menus
            }
            break;
// other cases or sub-menus
    }
// action
    switch(action){
        case 1: // or some other value
// take action
            break;
// other cases
    }
}
```

## Terms & Conditions

### **Quality Assurance**

A-WIT has stringent quality control procedures in place to insure the best quality products.

### **90-Day Limited Warranty**

A-WIT Technologies, Inc warrants its products against defects in materials and workmanship for a period of 90 days. If you discover a defect, A-WIT Technologies, Inc. will, at its option, repair, replace, or refund the purchase price. After 90 days, products can still be sent in for repair or replacement, but there will be a \$10.00USD minimum inspection/labor/repair fee (not including return shipping and handling charges).

### **14-Day Money-Back Guarantee**

If, within 14 days of having received your product, you find that it does not suit your needs, you may return it for a refund. A-WIT will refund the purchase price of the product in the form of a check, excluding shipping/handling costs, once the product is received. This refund does not apply if the product has been altered or damaged. If you decide to return the products after the 14-day evaluation period, a 20% restocking fee will be charged against a credit.

### **Disclaimer**

Warranty does not apply if the product has been altered, modified, or damaged. A-WIT makes no other warranty of any kind, expressed or implied, including any warranty of merchantability, fitness of the product for any particular purpose even if that purpose is known to A-WIT, or any warranty relating to patents, trademarks, copyrights or other intellectual property. A-WIT shall not be liable for any injury, loss, damage, or loss of profits resulting from the handling or use of the product shipped.

### **How to Return a Product**

When returning, you must first e-mail [sales@a-wit.com](mailto:sales@a-wit.com) for a Return Merchandise Authorization number. No packages will be accepted without the RMA number clearly marked on the outside of the package. After inspecting and testing, we will return your product, or its replacement using the same shipping method used to ship the product to A-WIT within 30 days. In your package, please include a daytime telephone number and a brief explanation of the problem.

Please contact our Sales Department at [sales@a-wit.com](mailto:sales@a-wit.com) if you have any questions regarding our warranty policy or if you are requesting an RMA number.

