Volume

1

A-WIT TECHNOLOGIES INC.

… a passion for execution …

CS47100X Bluetooth Reference Guide Manual

Version 1.2

A-WIT TECHNOLOGIES INC.

CS47100X Bluetooth Reference Manual

Phone (800) 985-AWIT • Fax (800) 985-2948

Table of Contents

Chapter

1

Introduction to the CS47100X Bluetooth Transceivers

The CS47100X Bluetooth transceivers are module devices for wireless serial communications using Bluetooth technology that is an internationally recognized standard for short range wireless communications. These modules can communicate with other Bluetooth devices that support the Serial Port Profile.

The CS471000 has a communication range of 30 m, and the CS471001 has a communication range up to 100 m, so this provides flexibility of suitability for different applications. The CS47100X devices deliver better quality of communication than a standard RS232 cables.

The CS47100X have a compact design and can be placed conveniently into devices, equipment, or breadboards.

The CS47100X support FHSS (Frequency Hopping Spread Spectrum), which is a technique, native to Bluetooth that allows the CS47100X to minimize radio interference while decreasing the likelihood of over-air hijacking. The CS47100X also support authentication and Bluetooth data encryption.

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 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 CS471000 Bluetooth Transceiver

The CS471000 Bluetooth module provides a very easy and low-cost solution for sending and receiving data between C Stamps or between a C stamp and a PC using Bluetooth connectivity. The module is compatible with the C Stamp microcomputer's supplies and signal levels. Controlling the module is made easy with A-WIT's supplied software commands BLUECMD, BLUEIN, and BLUEOUT that use the serial capability of the C Stamp. This simple three command interface is all that is required to control the Bluetooth module.

Technical Specifications (CS471000)

- Input/Output Interface: UART
- Compliant Bluetooth Specification v1.2
- Improved AFH (Adaptive Frequency Hoping)
- Fast connection
- Transmit Power: Max. +4dBm
- Receiving Sensitivity: -80dBm (0.1% BER)
- Compact size: 18 x 20 x 11.7 (mm)
- Provides transparent RS232 serial cable replacement
- Supports Bluetooth Serial Port Profile
- Interoperability with PDA, laptops etc.
- Built-in On-board antenna included
- Working distance (In an open field): Class 2, Nom. 30 meters
- Operating temperature: -10 °C to 55 °C
- Data Rate: 9,600 baud
- Breadboard-friendly
- Compatible with C Stamp microcomputer

Introduction to the CS471001 Bluetooth Transceiver

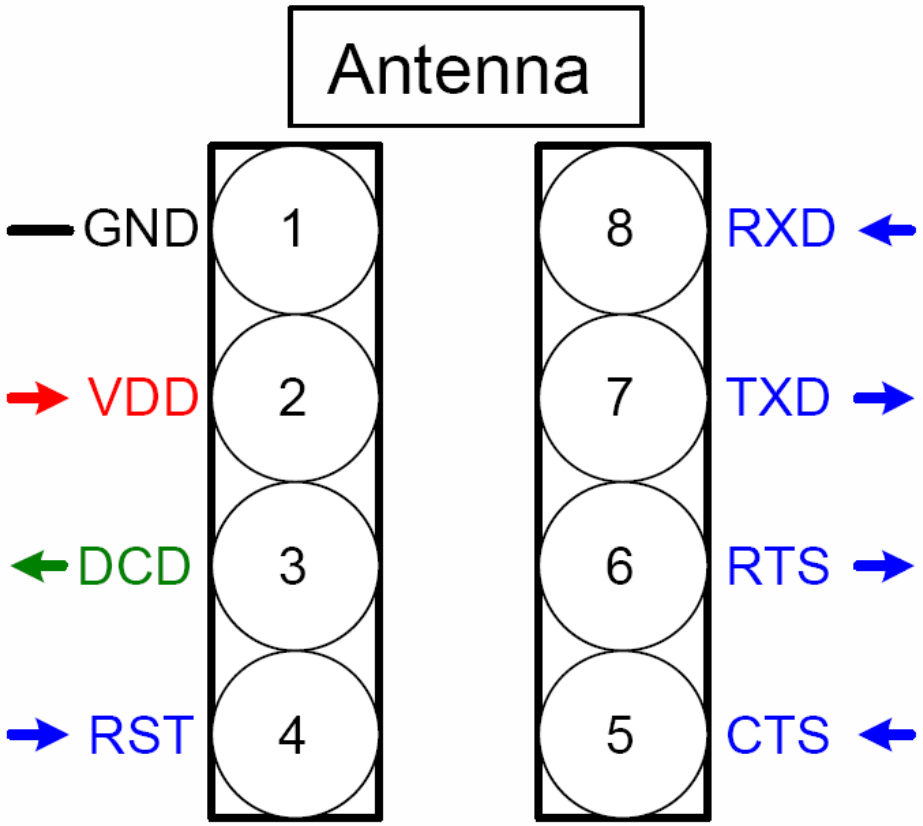
The CS471001 Bluetooth module provides a very easy and low-cost solution for sending and receiving data between C Stamps or between a C stamp and a PC using Bluetooth connectivity. The module is compatible with the C Stamp microcomputer's supplies and signal levels. Controlling the module is made easy with A-WIT's supplied software commands BLUECMD, BLUEIN, and BLUEOUT that use the serial capability of the C Stamp. This simple three command interface is all that is required to control the Bluetooth module.

Technical Specifications (CS471001)

- Input/Output Interface: UART
- Compliant Bluetooth Specification v1.2
- Improved AFH (Adaptive Frequency Hoping)
- Fast connection
- Transmit Power: Max. +18dBm
- Receiving Sensitivity: -88dBm (0.1% BER)
- Compact size: 27.5 x 27.7 x 14 (mm)
- Provides transparent RS232 serial cable replacement
- Supports Bluetooth Serial Port Profile
- Interoperability with PDA, laptops etc.
- Built-in On-board Chip antenna included
- Working distance (In an open field): Class 1, Nom. 100 meters
- Pairing Capability
- Operating temperature: -10 °C to 55 °C
- Data Rate: 9,600 baud
- Compatible with C Stamp microcomputer

General Information

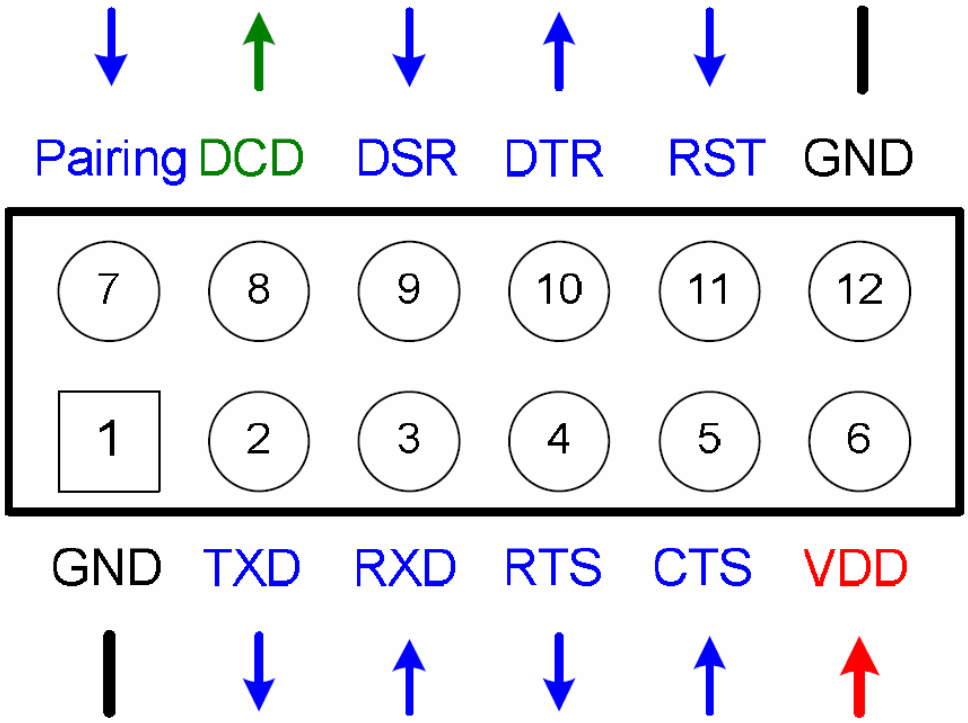
The following figure and table describe the pin assignment of the CS471000.



CS471000 Pin-Out Connectivity		
Pin	***Name***	***Description***
1	GND	Power Ground
2	VDD	DC Input (3.0 – 3.3 V)
3	Status	Bluetooth Connect Detect Output (active Low)
4	RST	Reset Input (Active Low)
5	CTS	Clear to Send Input

CS471000 Pin-Out Connectivity		
Pin	***Name***	***Description***
6	RTS	Ready to Send Output
7	TxD	Data Output
8	RxD	Data Input

The following figure and table describe the pin assignment of the CS471001.

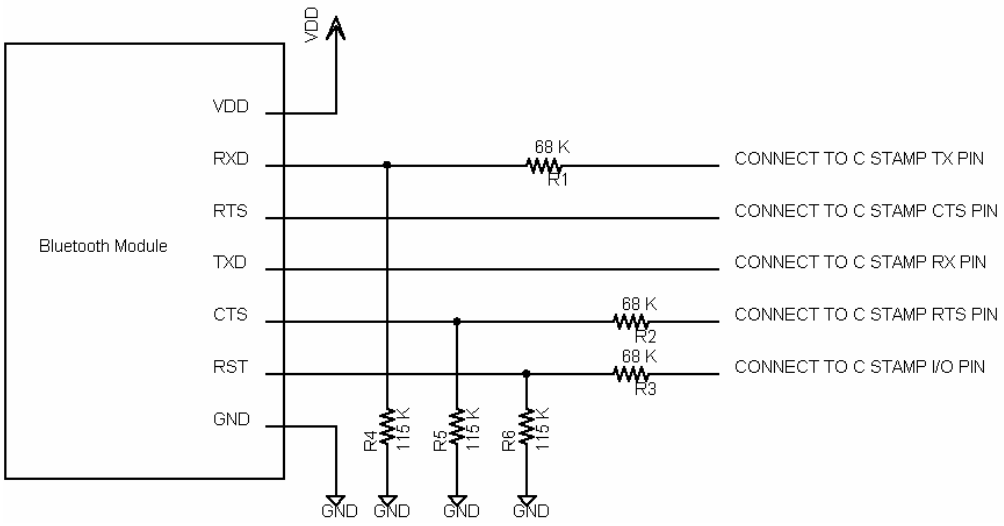


The figure above is a top view (on top of the connector), holding the module with the antenna on top and the connector (on the other side) at the bottom.

CS471001 Pin-Out Connectivity		
Pin	***Name***	***Description***

CS471001 Pin-Out Connectivity		
Pin	***Name***	***Description***
1	GND	Power Ground
2	TxD	Data Output
3	RxD	Data Input
4	RTS	Ready to Send Output
5	CTS	Clear to Send Input
6	VDD	DC Input (3.0 – 3.3 V)
7	Pairing	Pairing Input (Active Low)
8	Status	Bluetooth Connect Detect Output (active Low)
9	DSR	Data Set Ready Input
10	DTR	Data Terminal Ready Output
11	RST	Reset Input (Active Low)
12	GND	Power Ground

The figure below shows the minimal pin-out connectivity of any of the CS47100X Bluetooth devices to the C Stamp. The TX and RX connections to the C Stamp can be to any I/O pins, except the built-in asynchronous serial transmitter (Pin 24) and receiver (Pin 25), and the ATN pin (Pin 26).



Note that the VDD terminal in the figure above is not the same as the VDD voltages for the C Stamp or the Board of Learning (BOL). The C Stamp and the BOL use 5V as their supply level, while the Bluetooth modules use a VDD of 3.3V. You can use a Voltage Regulator, such as the A-WIT Part Number CS456000, to step down a 5V supply to a 3.3V level.

Operating Modes

The CS47100X has the following Bluetooth connection schemes.

A Bluetooth device can play roles as a master or as a slave. A master tries to connect itself to other Bluetooth devices, and a slave waits to be connected to from other Bluetooth devices. A Bluetooth connection is always made by a pair of master and slave devices. A slave can be in two modes, Inquiry Scan or Page. In Inquiry Scan mode, the slave is waiting for a packet of inquiry from another Bluetooth device; and in Page Scan mode, the slave is waiting for a packet of connection from another Bluetooth device. Every Bluetooth device has a unique address called BD (Bluetooth Device) address, which is comprised of 12 hexadecimal digits.

The CS47100X Bluetooth modules have 4 modes of operation as shown in the table below.

Name	***Description***
Mode 0	In this mode, there is no response when power on or reset is applied, and the CS47100X is just waiting for command input. Neither master nor slave role is assigned to the CS47100X in Mode 0. The user can

Name	*Description*
	change the configuration parameters of the CS47100X in this mode. The CS47100X comes in Mode 0 from the factory.
Mode 1	The CS47100X try to connect the last connected Bluetooth device. The CS47100X in Mode1 is to be a master and tries to connect to the last connected Bluetooth device. The CS47100X always stores the BD address of the Bluetooth device to which it was connected last. When the CS47100X is initially used, there is no BD address stored, so the CS47100X can be in Mode 1 only after a successful connection has been made to another Bluetooth device. Once in Mode 1, the CS47100X will try to connect automatically to the last connected Bluetooth device whenever the unit is powered on or reset. When the CS47100X is in Mode 1, it cannot be discovered or connected to by other Bluetooth devices.
Mode 2	The CS47100X is waiting for a connection from the last connected Bluetooth device. The CS47100X in Mode 2 is to be a slave and waits for a connection only from the last connected Bluetooth device. Just like Mode 1, if there is no BD address stored in the CS47100X, the device cannot be in Mode 2. Once in Mode 2, the CS47100X will wait for a connection from the last connected Bluetooth device whenever the unit is powered on or reset. When the CS47100X is in Mode 2, it cannot be discovered or connected to by other Bluetooth devices, other than the one that was last connected.
Mode 3	The CS47100X is waiting for a connection from any other Bluetooth devices. In Mode 3, the CS47100X is discoverable and can be connected to by any other Bluetooth device.

RST Signal

The RST signal can be used for setting the CS47100X to their factory default state. RST should be at 0 V for at least 1 second for the reset to factory mode to occur. Otherwise; the CS47100X can be reset normally by holding the RST signal LOW for half a second.

Pairing Signal (CS471001 Only)

The CS471001 provides a pairing signal input for instant configuration and automatic connection of two CS471001's. In this example, we will name the two CS471001's as CS471001A and CS471001B.

Step 1. Turn off all the nearby CS471001's.

Step 2. Turn on CS471001A and the CS471001B and reset both of them by using RST signals.

Step 3. Set the pairing signal of CS471001A to a LOW state and hold the signal for 2 seconds.

Step 4. Set the pairing signal of CS471001B to a LOW state and hold the signal for 2 seconds. Set the pairing signal of CS471001B to a HIGH state and hold the signal for 2 seconds. Now, set the pairing signal of CS471001B to a LOW state and hold it for 2 seconds.

Step 5. Wait for CS471001A and CS471001B to connect to each other. It may take about 10 seconds to make a connection. If there are many Bluetooth devices nearby, the connection time may increase.

Step 6. At this point your pair of CS471001's is configured to make automatic connection to each other. You can now use this pair of CS471001's like virtual serial cable.

The Pairing process using the Pairing Signal is also shown in the table below.

CS471001A	***Status***	***Pairing Signal***	***CS471001B***	***Status***	***Pairing Signal***
1. Reset	Mode 0	HIGH	1. Reset	Mode 0	HIGH
2. Drop pairing signal	Mode 3	LOW	2. Drop pairing signal	Mode 3	LOW
3. Restore pairing signal	Mode 3	HIGH	3. Restore pairing signal	Mode 3	HIGH
			4. Drop pairing signal	Mode 1	LOW
			5. Restore pairing signal	Mode 1	HIGH

CS471001A	***Status***	***Pairing Signal***	***CS471001B***	***Status***	***Pairing Signal***
6. Connected	Slave	HIGH	6. Connected	Master	HIGH

FCC Notice

The CS47100X Bluetooth modules comply with Part 15 of the FCC rules and regulations.

Chapter

2

Getting Started

This chapter is a quick start guide to using the CS47100X Bluetooth Transceiver Module with the C Stamp. This assumes you have two C Stamps with appropriate connection kits or development boards with the RESET and START circuitries, at least 1 LED connected to each C Stamp, two CS471000's Bluetooth Transceiver Modules properly connected to the C Stamps, and an optional utility input button circuit connected to the first 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

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

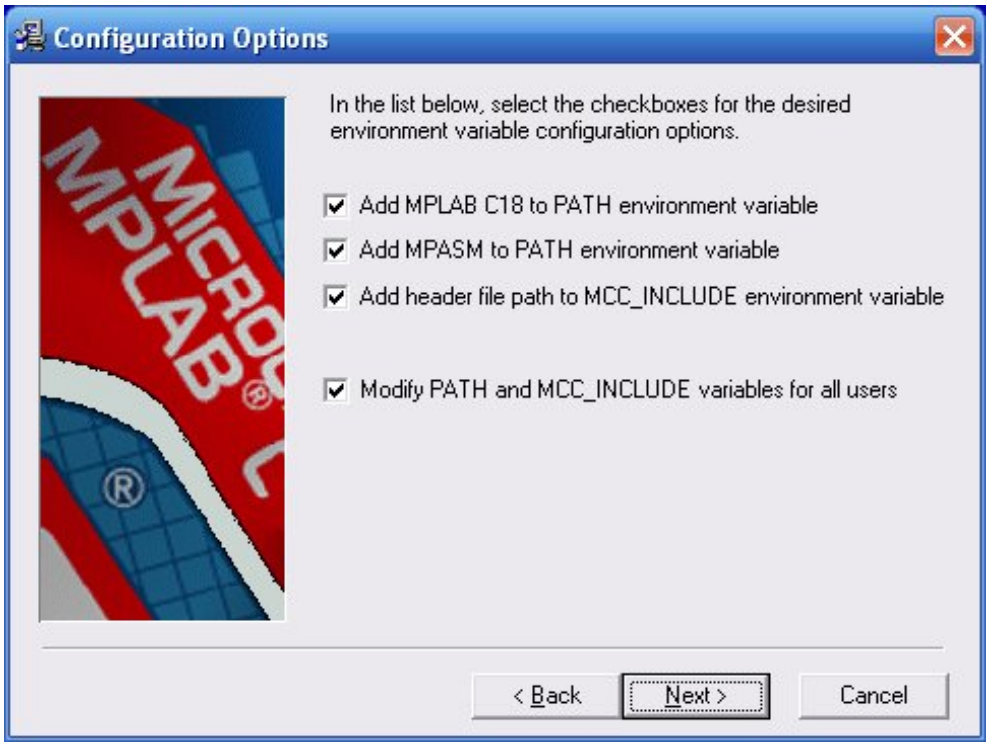
If possible, please check the C Stamp website www.c-stamp.com under SUPPORT for any updates to documentation, changes, or notices that may have become available since your Installation CD was produced. If you continue to have any issues for which a solution is not found in the aforementioned website,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

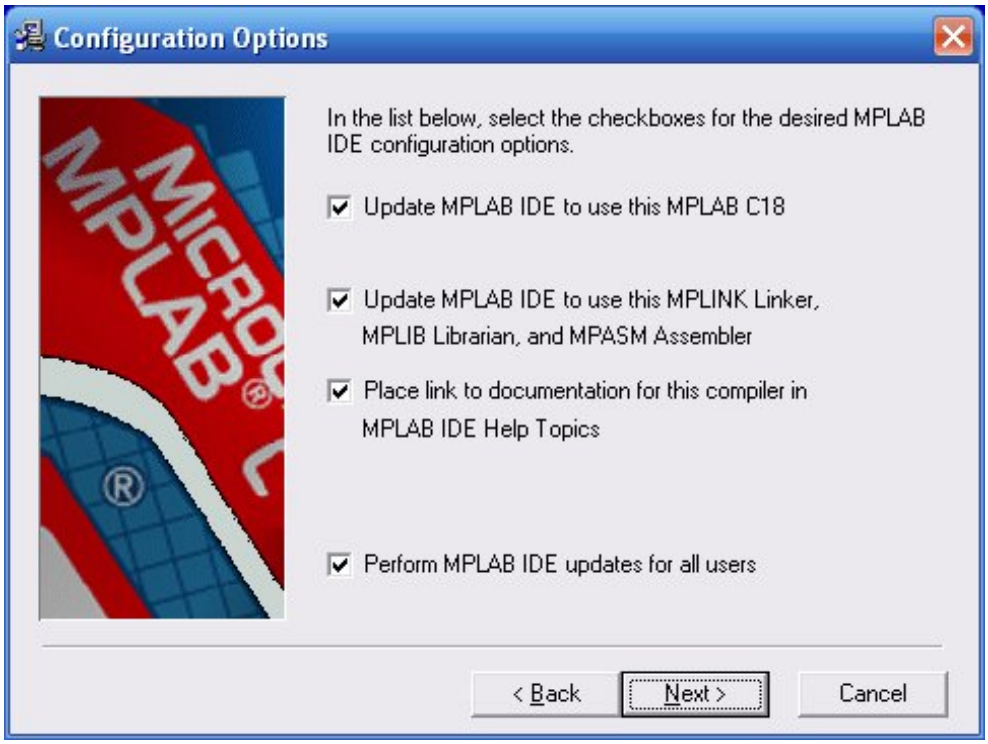
Installing the Microchip MPLAB and C Compiler Software

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 5th and 6th 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.


Configuration Options
In the list below, select the checkboxes for the desired environment variable configuration options.
Add MPLAB C18 to PATH environment variable
Add MPASM to PATH environment variable
Add header file path to MCC_INCLUDE environment variable
Modify PATH and MCC_INCLUDE variables for all users
< Back
Next >
Cancel


Configuration Options
In the list below, select the checkboxes for the desired MPLAB IDE configuration options.
Update MPLAB IDE to use this MPLAB C18
Update MPLAB IDE to use this MPLINK Linker, MPLIB Librarian, and MPASM Assembler
Place link to documentation for this compiler in MPLAB IDE Help Topics
Perform MPLAB IDE updates for all users
< Back
Next >
Cancel

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

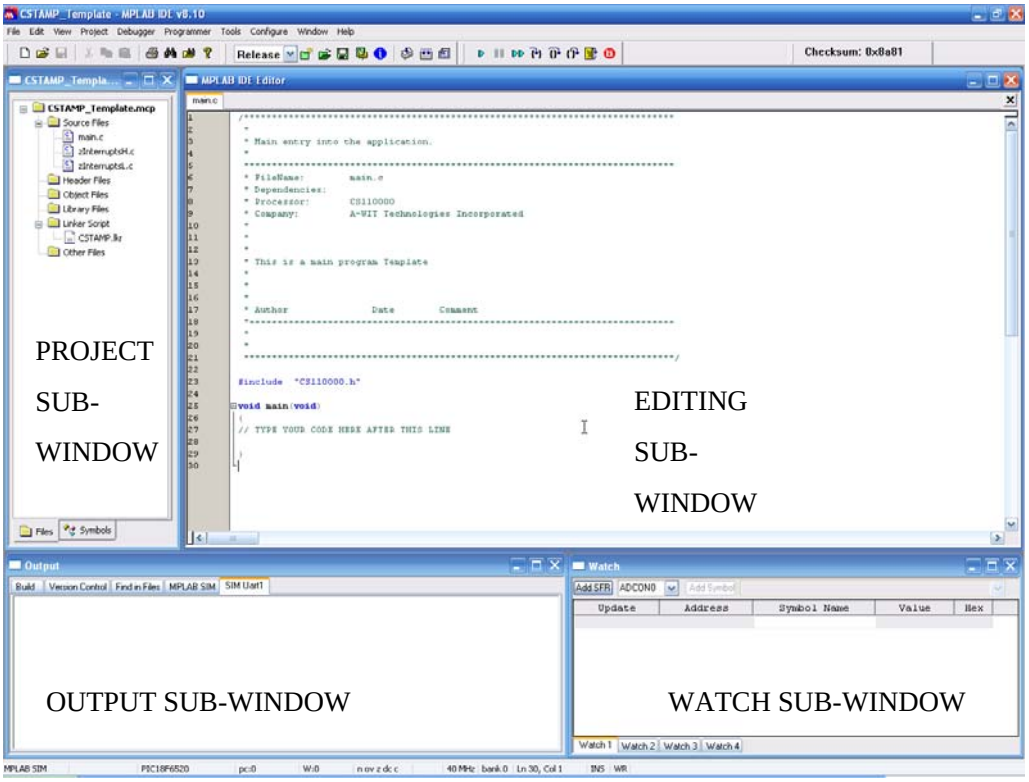
Copy the DOCS directory from the C Stamp Installation CD to your C:\A-WIT directory. This directory contains all the C Stamp related documentation in PDF format.

Creating your Bluetooth Transceiver Programs

For each program follow the following instructions. Create a directory where you want to have all the files for your program; for example, BTA_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 "BTA_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 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. Each program assumes that you have the Bluetooth Transceiver Module connected to Pin 1 for transmission into the Bluetooth Module, Pin 2 connected to the RST pin in the Bluetooth module (this is the C Stamp CTS pin), Pin 3 being used for reception from the Bluetooth Module, Pin 4 connected to the CTS pin in the Bluetooth module (this is the C Stamp RST pin), the RST pin in each Bluetooth Transceiver Module in each setup connected to Pin 8, the LED in each setup connected to Pin 46, and the utility button in the first setup connected to Pin 37. After both setups have been STARTed, they will setup a communication link between each other, in which the first setup is the master, and the second setup is the slave. The first program monitors the button, and if the button is pushed, it will send a signal (a byte with a value of 1) to the second setup. The second setup will monitor transmission from the first setup, and if a byte with the value of 1 is received, it will light the LED and it will send a byte with a value of 1 back to the first setup signaling that the "turn on the LED" command was received. After this action is taken, the program will end execution. The first program will also monitor transmission from the second to receive the acknowledgement that the "turn on the LED" command was received. When this handshake is received, it will turn on its own LED, and end execution. The programs stay in that mode indefinitely until you restart each of them by pushing and releasing the RESET button while holding the START button and then letting go of the latter. In each setup, signaling that a Bluetooth connection has been established is done by lighting the LED connected to Pin 45.

FIRST PROGRAM SEGMENT

```
// Declare some necessary variables
  BIT button_pushed;  // For button
  BYTE data[] = {1};
  WORD Wparms[4] = {3, 0, 0, 0};
  BYTE Pins[] = {1, 2, 3, 4};
  WORD Wresps[3];

// Power off LEDs connected to pins 46 and 45
  STPIND(46, LOW);
  STPIND(45, LOW);

// reset BT module
  STPIND(8, HIGH);
  PAUSE(1000);
  STPIND(8, LOW);
  PAUSE(500);
  STPIND(3, HIGH);
  PAUSE(1000);

// setup communication link
  BLUECMD_CS47100X(BT_MODE, Wparms, Pins, 0, 0);
  BLUECMD_CS47100X(BT_SCAN, Wparms, Pins, Wresps, 0);

// Power on LED connected to pin 45
  STPIND(45, HIGH);

// Wait for button connected to pin 37 to be pushed
  button_pushed = FALSE;
  while(!button_pushed){
    button_pushed = BUTTON(37, LOW, HIGH, 5);
  }

// Send byte of 1
  BLUEOUT_CS47100X (data, 1, Pins);

// Reset the data
  data[0] = 0;

// Get handshake
  BLUEIN_CS47100X (0, data, 1, Pins);

// Turn on LED if communication was successful
  if(data[0] == ONE) STPIND(46, HIGH);

  STOP();
```

SECOND PROGRAM SEGMENT

```
// Declare some necessary variables
  BYTE data[] = {0};
  WORD Wparms[4] = {3, 0, 0, 0};
  BYTE Pins[] = {1, 2, 3, 4};
  WORD Wresps[3];

// Power off LEDs connected to pins 46 and 45
  STPIND(46, LOW);
  STPIND(45, LOW);

// reset BT module
  STPIND(8, HIGH);
  PAUSE(1000);
  STPIND(8, LOW);
  PAUSE(500);
  STPIND(3, HIGH);
  PAUSE(1000);

// setup communication link
  BLUECMD_CS47100X(BT_MODE, Wparms, Pins, 0, 0);
  BLUECMD_CS47100X(BT_SCAN, Wparms, Pins, Wresps, 0);

// Power on LED connected to pin 45
  STPIND(45, HIGH);

// Get data
  BLUEIN_CS47100X (0, data, 1, 2);

// If data is 1, turn on LED and send acknowledgement
  if(data[0] == ONE){
    STPIND(46, HIGH);
// Send byte of 1
    BLUEOUT_CS47100X (data, 1, Pins);
  }
  else{
// send back a 0
    data[0] = ZERO;
    BLUEOUT_CS47100X (data, 1, Pins);
  }

  STOP();
```

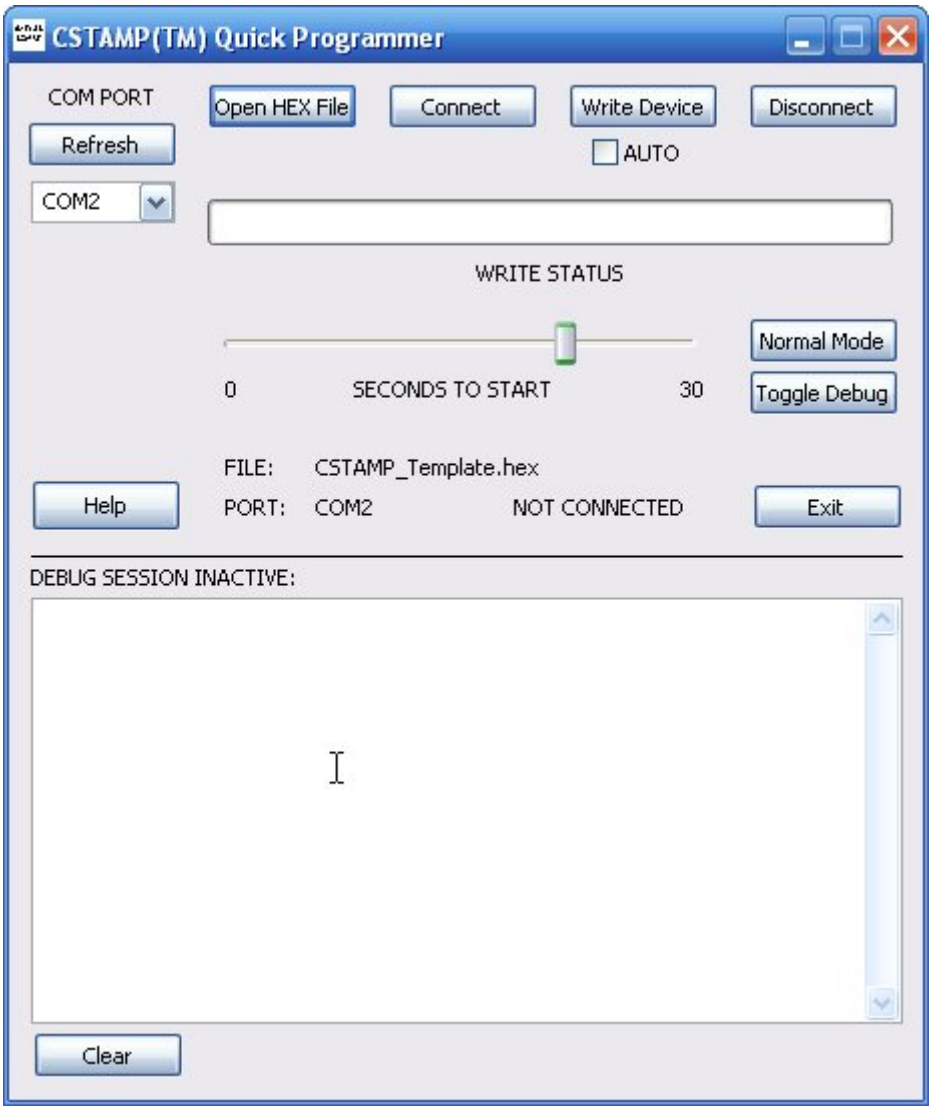
Save your program from the "File" menu or by clicking on the appropriate icon in the tool bar. Then "Build All" from the "Project" menu or from the tool bar.

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 BTA_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.

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 are 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 started 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.

Chapter 3

Accessory Specific Functions and Commands Reference

This chapter describes the functions and commands that are specific to the software support of different types of accessories that are available from A-WIT Technologies to complement the function and projects developed with the C Stamp. The user should consult the manual for a specific accessory for full information on connectivity and usage.

BLUECMD_CS47100X

```
BIT BLUECMD_CS47100X(BYTE command, WORD Wparameters[],
                     BYTE Pins[], WORD Wresponses[],
                     BYTE Sparameter[]);
```

The **BLUECMD** function sends a command to the CS47100X Bluetooth Module according to the tables below. If the function is successful, it returns **TRUE**; otherwise, it returns **ZERO**. This could mean that there was an error in the arguments of the function or some other problem.

command is a variable/constant/expression (1 – 17) indicating the BLUETOOTH command to send.

Wparameters is a 4 WORD array that is used by different commands. This parameter is not used by all the commands. If the parameter is not used by a command, any value can be passed to the function (e.g. **ZERO**), and it will be ignored. However; something always must be passed in all the arguments. If the usage of the parameter is not specified in the tables below, then it is not used and ignored by the function.

Pins is an array that defines the C Stamp pin numbers that connect to the Bluetooth module for serial communication. These pins are defined from the perspective of the

C Stamp in this order: Tx, CTS, Rx, and RTS. These pins, in turn connect to the following pins of the Bluetooth module respectively: RXD, RTS, TXD, and CTS. The C Stamp Tx and RTS pins will be set to output mode, and Rx and CTS will be set to input mode.

Wresponses is a 3 WORD array that is used by different commands. This array is not used by all the commands. If the array is not used by a command, any value can be passed to the function (e.g. **ZERO**), and it will be ignored. However; something always must be passed in all the arguments. If the usage of the array is not specified in the tables below, then it is not used and ignored by the function.

Sparameter is a string array of 16 alpha-numeric characters that is used by different commands. This parameter is not used by all the commands. If the parameter is not used by a command, any value can be passed to the function (e.g. **ZERO**), and it will be ignored. However; something always must be passed in all the arguments. If the usage of the parameter is not specified in the tables below, then it is not used and ignored by the function.

Command Categories			
Command Category		***Value***	**command** ***Symbol***
RESET		1	**BT_SWRST**
		2	**BT_HWRST**
BLUETOOTH	Mode	3	**BT_MODE**
	Status	4	**BT_CANCEL**
		5	**BT_SCAN**
		6	**BT_SCANT**
		7	**BT_SCANA**
	Connection	8	**BT_CLAST**
		9	**BT_CA**
		10	**BT_REL**
	Information	11	**BT_ALAST**
		12	**BT_SIGSTR**

Command Categories			
Command Category		*Value*	**command** *Symbol*
	Security	13	**BT_KEY**
		14	**BT_STKEY**
		15	**BT_STAUTH**
	Miscellaneous	16	**BT_STLP**
		17	**BT_NAME**

command *Symbol*	*Value*	*Command*	*Description*
BT_SWRST	1	Software Reset	This has the same effects as power cycling the unit. This command disconnects any connected Bluetooth devices, and stops ongoing tasks. After rebooting, the status will be decided by the preset operation mode.
BT_HWRST	2	Hardware Reset	All parameters are initialized to factory defaults.
BT_MODE	3	Set Operating Mode	Operating Mode is set to the one given by **Wparameters[0]**. **Wparameters[0]** = 0: MODE 0, (Default) **Wparameters[0]** = 1: MODE 1 **Wparameters[0]** = 2:

command *Symbol*	*Value*	*Command*	*Description*
			MODE 2 **Wparameters[0]** = 3: MODE 3
BT_CANCEL	4	Terminate the Current Task	This terminates the current task being executed, such as Inquiry scan and Page scan.
BT_SCAN	5	Wait for Inquiry and Connection From Other Bluetooth Devices	This allows Inquiry and Connection from the other Bluetooth devices. When connection is made the BD address of the connected device is returned in the **Wresponses** array as follows: **Wresponses[0]** = Left most 4 digits of the BD Address **Wresponses[1]** = Middle 4 digits of the BD Address **Wresponses[2]** = Right most 4 digits of the BD Address
BT_SCANT	6	Wait for Inquiry and Connection From Other Bluetooth Devices for a Given Duration	The module waits for **Wparameters[1]** seconds for Inquiry and Connection from other Bluetooth devices. If the **Wparameters[1]** is 0, the module waits forever. **Wparameters[0]** = 1: Allows Inquiry Scan **Wparameters[0]** = 2:

command *Symbol*	*Value*	*Command*	*Description*
			Allows Page Scan **Wparameters[0]** = 3: Allows both Inquiry and Page Scan When connection is made the BD address of the connected device is returned in the **Wresponses** array as follows: **Wresponses[0]** = Left most 4 digits of the BD Address **Wresponses[1]** = Middle 4 digits of the BD Address **Wresponses[2]** = Right most 4 digits of the BD Address
BT_SCANA	7	Wait for Connection by the Bluetooth Device with the Given Bluetooth Device (BD) Address	The module waits for **Wparameters[3]** seconds to be connected to by the Bluetooth device with the given BD address. If **Wparameters[3]** is 0, the module waits forever. **Wparameters[0]** = Left most 4 digits of the BD Address **Wparameters[1]** = Middle 4 digits of the BD Address **Wparameters[2]** = Right most 4 digits of the BD Address

command *Symbol*	*Value*	*Command*	*Description*
BT_CLAST	8	Connect to the Last Connected Bluetooth Device	The CS47100X devices save the BD Address of the Bluetooth device that was last connected to it. When connection is made the BD address of the connected device is returned in the **Wresponses** array as follows: **Wresponses[0]** = Left most 4 digits of the BD Address **Wresponses[1]** = Middle 4 digits of the BD Address **Wresponses[2]** = Right most 4 digits of the BD Address
BT_CA	9	Connect to a Specific Bluetooth Device with a Given BD Address	The CS47100X attempts to connect to the Bluetooth device with the given BD address. To make a successful connection, the Bluetooth device must be in Page scan mode. This attempt continues for 5 minutes. **Wparameters[0]** = Left most 4 digits of the BD Address **Wparameters[1]** = Middle 4 digits of the BD Address **Wparameters[2]** = Right most 4 digits of the BD Address
BT_REL	10	Release the Current Connection	The current Bluetooth connection is disconnected. It takes about 30 seconds to detect an abnormal

command *Symbol*	*Value*	*Command*	*Description*
			disconnection such as power off or moving out of service range.
BT_ALAST	11	Returns the BD Address of the Last Connected Device	The CS47100X devices return the BD Address of the Bluetooth device that was last connected to it. This is returned in the **Wresponses** array. **Wresponses[0]** = Left most 4 digits of the BD Address **Wresponses[1]** = Middle 4 digits of the BD Address **Wresponses[2]** = Right most 4 digits of the BD Address
BT_SIGSTR	12	Test Signal Strength	When a Bluetooth connection is established, you can use this command to query the module for the signal strength. This is returned in the **Wresponses** array. **Wresponses[0]** = LinkQuality with range [0, 255]. Higher is better. **Wresponses[1]** = RSSI (interference) with range [0, 255]. Lower is better. If the LinkQuality is close to 255, and the RSSI is close to 0, it implies that the signal strength is good.
BT_KEY	13	Change Pin Code	Pin code is a string in the **Sparameter** array, which allows up to 16 alpha-numeric characters. Based

command *Symbol*	*Value*	*Command*	*Description*
			on this pin code, the CS47100X generates a link key which is used in the actual authentication process. The default code is "1234".
BT_STKEY	14	Set Generation of Link Key Every Time a Connection Is Made	If **Wparameters[0]** is set to 1, the CS47100X asks for the pin code every time a connection is made. This can be used to increase security. If **Wparameters[0]** is set to 0, this feature is deactivated.
BT_STAUTH	15	Set Authentication and Data Encryption	Authentication is set with **Wparameters[0]**. **Wparameters[0]** = 0: Inactive (Default) **Wparameters[0]** = 1: Active Encryption is set with **Wparameters[1]**. **Wparameters[1]** = 0: Inactive (Default) **Wparameters[1]** = 1: Active
BT_STLP	16	Set Low Power Mode	If **Wparameters[0]** is set to 1, Low Power Mode is enabled. If **Wparameters[0]** is set to 0, this feature is deactivated. When activated, and while data is not

command *Symbol*	*Value*	*Command*	*Description*
			being transmitted or received, the CS47100X goes into low power mode to save power. It takes a few seconds to wake the CS47100X from low power mode.
BT_NAME	17	Change Device Name	The CS47100X can have a user friendly name for easy identification. The name is allowed to be up to 30 alpha-numeric characters, and is passed in the **Sparameter** character array. The default is "PSDv3b-445566".

BLUEIN_CS47100X

NIBBLE BLUEIN_CS47100X(WORD timeout, BYTE buffer[], BYTE Lbuffer, BYTE Pins[]);

The **BLUEIN** function receives asynchronous serial data from the CS47100X Bluetooth Module via any I/O pin, except the built-in asynchronous serial transmitter (Pin 24) and receiver (Pin 25), and the ATN pin (Pin 26).

timeout is a variable/constant/expression (0 – 65535) that tells **BLUEIN** how long to wait in mS for incoming data. If data does not arrive in time or if the buffer gets full before the timeout condition occurs, the function will return with the appropriate return code. The value of 0 is special; it indicates that the function will wait until it receives enough data to fill the buffer before it returns. The usage of timeouts applies to each data byte being received.

buffer is the name of the array of **BYTES** that the function will use as storage to return the received data to the calling function. This can be an array of length equals to 1 byte, and the maximum length of the buffer is 255 bytes.

Lbuffer is the maximum number of bytes that can be placed in **buffer**. If fewer bytes are received than what can be placed in the buffer, the buffer gets filled from the low to high direction of its index address space (i.e. from 0 to **Lbuffer**-1).

Pins is an array that defines the C Stamp pin numbers that connect to the Bluetooth module for serial communication. These pins are defined from the perspective of the

C Stamp in this order: Tx, CTS, Rx, and RTS. These pins, in turn connect to the following pins of the Bluetooth module respectively: RXD, RTS, TXD, and CTS. The C Stamp Tx and RTS pins will be set to output mode, and Rx and CTS will be set to input mode.

On exit, the function returns one of the following exit codes.

return code	*value*	*Meaning*
BTIN_ARGERR	0	There was at least one error in the arguments of the function. Function did not execute.
BTIN_BUFULL	1	Function returned with the buffer full of data.
BTIN_TIMOUT	2	The function exited after a timeout condition occurred. The highest index of buffer (Lbuffer-1) contains the index of the last byte that was received successfully before the timeout occurred. If this location contains the value 0xFF, this indicates that no data was received.
BTIN_OVRERR	3	An overrun error occurred. The highest index of buffer (**Lbuffer**-1) contains the index of the last byte that was received successfully before the error occurred. If this location contains the value 0xFF, this indicates that no data was received.

BLUEOUT_CS47100X

```
BIT BLUEOUT_CS47100X(BYTE buffer[], BYTE Lbuffer,
                     BYTE Pins[]);
```

The **BLUEOUT** function transmits asynchronous serial data to the CS47100X Bluetooth Module via any I/O pin, except the built-in asynchronous serial transmitter (Pin 24) and receiver (Pin 25), and the ATN pin (Pin 26).

buffer is the name of the array of **BYTES** that the function will send. This can be an array of length equals to 1 byte, and the maximum length of the buffer is 255 bytes.

Lbuffer is the maximum number of storage bytes to be sent in the buffer. The buffer gets processed for transmission from the low to high direction of its index address space (i.e. from 0 to **Lbuffer**-1).

Pins is an array that defines the C Stamp pin numbers that connect to the Bluetooth module for serial communication. These pins are defined from the perspective of the C Stamp in this order: Tx, CTS, Rx, and RTS. These pins, in turn connect to the following pins of the Bluetooth module respectively: RXD, RTS, TXD, and CTS. The C Stamp Tx and RTS pins will be set to output mode, and Rx and CTS will be set to input mode.

On exit, the function returns one of the following exit codes.

return code	*value*	*Meaning*
BT_ARGERR	0	There was at least one error in the arguments of the function. Function did not execute.
BT_BUFEMP	1	Function returned after transmitting all data in the buffer.

Terms and 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 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 if you have any questions regarding our warranty policy or if you are requesting an RMA number.