Web Site: www.parallax.com
Forums: forums.parallax.com
Sales: sales@parallax.com
Technical: support@parallax.com

Office: (916) 624-8333
Fax: (916) 624-8003
Sales: (888) 512-1024
Tech Support: (888) 997-8267

Easy Bluetooth (#30085)

The Easy Bluetooth is a RoboTech RBT-001 Bluetooth serial module with an adapter specifically designed to be used with the Parallax Board of Education® AppMod Header or breadboard. In addition, the Easy Bluetooth can plug into any 0.1 inch spacing development platform which allows it to be breadboard friendly, and also easy to implement into a soldering application. The module has two parts, the RBT-001 module and the SIP with voltage regulator PCB. With the on-board regulator, the module can be connected to voltages higher than 3.3 VDC, such as the Board of Education regulated supply (+5 VDC) without worry of damaging the unit; while the RX and TX can utilize serial communication at CMOS and TTL levels.

Features

- 1.x & 2.0 Bluetooth Compliant
- Class 2 Operation (nominal range up to 30 meters)
- 10-pin SIP package for breadboard, perfboard, or Board of Education AppMod Header
- On-board regulator for safe operations across various voltages
- CMOS & TTL Compatible

Key Specifications

- Power requirements: 3.3 to 5.5 VDC
- Communication: UART Command/Data Port supports for up to 921.6k baud
- Operating temperature: +32 to +113 °F (0 to +45 °C)
- Dimensions: 1.40 x 1.79 x .49 in (34.41 x 45.65 x 12.51 mm)

Application Ideas

- Control a Boe-Bot via Bluetooth from a PC, Cell Phone, or another Bluetooth module
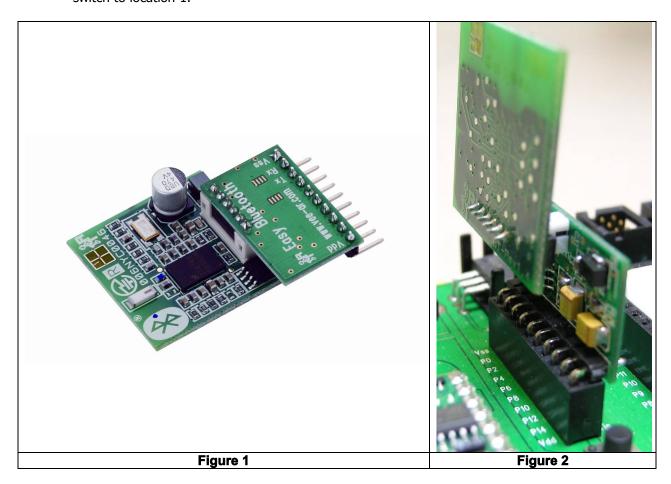- Communicate with a device or project wirelessly

Packing List

- RBT-001 Module
- 10-pin SIP with connector Module (pre-assembled with RTB-001)

Quick Start Circuit for the BASIC Stamp 2 and Board of Education

There are a few steps to take to install an Easy Bluetooth module, creating a Bluetooth connection on a PC operating with Windows XP, and finally writing a program for the BASIC Stamp® 2. Once you are done, you will have a working Easy Bluetooth module communicating with the PC via Bluetooth.

Installing the Easy Bluetooth Module

1. Carefully open the Easy Bluetooth package and check that the two small boards are properly plugged into each other as shown in Figure 1.

2. Now plug the Easy Bluetooth module into the AppMod header of the Board of Education: insert the module in the left row of the AppMod Header and notice there are labels on the module that indicate correct pin placement (Rx uses P0, Tx uses P2, Vss with Vss and Vdd with Vdd). Visually confirm the module is inserted correctly using Figure 2 before powering.

3. Connect a communication cable and power supply to the Board of Education and turn the power switch to location 1.

Figure 1	Figure 2

Creating a new Bluetooth Connection

1. Be sure that the Bluetooth dongle that you installed on your PC is working correctly and is turned on (see Bluetooth Dongle user manual for specific instructions to ensure proper operation).

    a. Open Control Panel, click "Printer and Other Hardware" and then click Bluetooth Devices. If in Classic View double click "Bluetooth Devices".

    b. Click "Add".

c. Check "My Device is set up and ready to be found" in the "Add Bluetooth Device Wizard" and click "Next".

d. Select "EasyBT" from the displayed Bluetooth devices and click "Next".

e. Select "Use the passkey found in the documentation" and enter the Passkey Code 0000 (zero four times) and click "Next".

f. Bluetooth Manager should display and designate an Outgoing and Incoming COM port for the Easy Bluetooth device: make a note of the COM ports because these are the ports you will use to communicate with the Easy Bluetooth.

g. Select "Finish" to complete the Bluetooth device configuration.

2. To close the "Bluetooth Devices" window click "OK".

A test for the BASIC Stamp 2 microcontroller

1. The Easy Bluetooth can communicate at numerous baud rates, and with many different microcontrollers. This example uses the AppMod Header on the Board of Education, the BASIC Stamp 2, at 9600 baud.

2. The easiest way to see in your code if a Bluetooth connection has been established is to wait for a byte to be received. You can copy the program below into the BASIC Stamp Editor to ensure a proper connection.

```
' {$STAMP BS2}
' {$PBASIC 2.5}

RX            CON   2      'Receive Pin
TX            CON   0      'Transmit Pin
Baud          CON   84     '9600 Baud

combyte            VAR Byte   'Communication Byte

DEBUG "Use This Screen for Display",CR

DO
  'Wait for a first byte indicating an active Bluetooth connection
  SERIN RX, Baud, [combyte]
  DEBUG combyte
LOOP
```

3. Download the program using Run => Run (ctrl + r) from the BASIC Stamp Editor

4. Keep Current DEBUG Screen open, and open another DEBUG Screen (ctrl +d) from the BASIC Stamp Editor, and select TX COM that the Bluetooth module (refer to COM ports during Bluetooth installation) is using; you should now have 2 DEBUG Terminal windows open. Click in the white area of the DEBUG window of DEBUG Terminal 2 and press any key; if all the steps were correctly done, the same key will be sent back to the DEBUG Terminal 1 window.

Resources and Downloads

You may download the RBT-001 user manual, example source codes, PC applications, and mobile phone application from www.parallax.com.

*Mobile phone must be compatible with the JSR-82 Java API; usually all Nokia and Sony Ericsson phone are compatible, but check users manual for compatibility. Configuration of mobile phone is not supported by Parallax Technical Support.

Device Information

Theory of Operation

Bluetooth is an open wireless protocol for exchanging data over short distances from fixed and mobile devices, creating Personal Area Networks (PANs). It was originally conceived as a wireless alternative to RS232 data cables. It can connect several devices, overcoming problems of synchronization.

There are various versions of Bluetooth communication and the Easy Bluetooth is compatible with versions 1.x and 2.0.

Precautions

The Easy Bluetooth uses the microwave radio frequency spectrum in the 2.4 GHz to 2.4835 GHz range. Maximum power output from this Bluetooth radio 2.5 mW; Class 2 devices are considered less of a potential hazard than mobile phones.
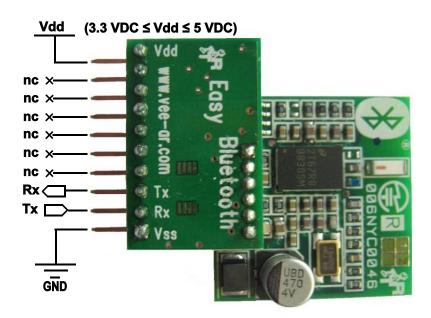
Pin Definitions and Ratings

Pin	Name	Function
1	Vdd	+3.3 to +5 VDC (+5.5 max voltage)
2	RX	Receive Serial Communication (CMOS & TTL Level Compatible)
3	TX	Transmit Serial Communication (CMOS & TTL Level Compatible)
4	NC	Not Connected
5	NC	Not Connected
6	NC	Not Connected
7	NC	Not Connected
8	NC	Not Connected
9	NC	Not Connected
10	Vss	Ground

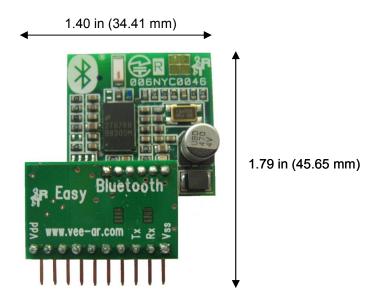## Communication Protocol

UART Command/Data Port supports up to 921.6k non-inverted baud.

Connection Diagram

Vdd (3.3 VDC ≤ Vdd ≤ 5 VDC)

nc ✕
nc ✕
nc ✕
nc ✕
nc ✕
nc ✕
Rx
Tx

GND

Module Dimensions

1.40 in (34.41 mm)

1.79 in (45.65 mm)

Example Application

This example application uses the RoboTech SRL GUI interface to control a Boe-Bot® robot via Bluetooth.

*A note to mention is that this application uses a standard operational Boe-Bot; so please have an operational Boe-Bot ready while completing this example. For help building a fully operational Boe-Bot, please refer to Robotics with the Boe-Bot Text.

To install the GUI software, complete the following steps:

1. Install the Easy Bluetooth PC software from the "Easy Bluetooth Software.zip" file with all the default and install locations as prompted by the installer. At the end of the installations you will find a new program folder "Easy Bluetooth" under the "Parallax Inc" main folder named with a sub folder named "BoeBot Remote Control", in your Start Menu.

2. Select "BoeBot Remote Control" application from the sub-folder listed in the previous step.

3. Click "Options" and select the COM port that was previously established with the Bluetooth configuration.

4. Download the program below using the BASIC Stamp Editor (ctrl + r) to the BASIC Stamp 2 and then download the program below to the BASIC Stamp 2 using Run => Run or (crtl + r).

5. Disconnect the Boe-Bot from PC

6. Click "Connect" in the BoeBot Remote Control application to gain control of the Boe-Bot.

BASIC Stamp® 2 Program

```
'' ========================================================================
'
'   File...... Easy Bluetooth.bs2
'   Purpose... Control a Boe-Bot with a GUI interface
'   Author.... Technical Support & RoboTech SRL
'   E-mail.... support@parallax.com
'   Started... April 1st 2009
'   Updated... N/A
'
'   {$STAMP BS2}
'   {$PBASIC 2.5}
'
' ========================================================================


' -----[ Program Description ]--------------------------------------------

' This program is what is loaded into the BASIC Stamp prior to using the GUI interface
' from RoboTech SLR. It allows a Boe-Bot to be controlled via a GUI interface.
'
' Controls are as followed:
'
'     Up Arrow     = Forward
'     Down Arrow   = Backwards
'     Left Arrow   = Left Turn
'     Right Arrow  = Right Turn
```

```
'     Space Bar   = Stop
'
' You can press and hold and release an arrow key for 2 seconds to access a double
' speed of each action. For example, Press Up Arrow, release to make the Boe-Bot move
' faster forward.

' -----[ I/O Definitions ]-------------------------------------------------

BT_RX           PIN    2            ' RX of the Easy Bluetooth
BT_TX           PIN    0            ' TX of the Easy Bluetooth
LED             PIN    5            ' Indicator LED for Bluetooth Connection

' -----[ Constants ]-------------------------------------------------------

BT_SPEED        CON    84           ' baud 9600 true UART

' -----[ Variables ]-------------------------------------------------------

tLeft           VAR    Word         ' Left Servo control pulse durations
tRight          VAR    Word         ' Right Servo control pulse durations
temp            VAR    Word         ' Temp variable

' Buffer array not declared as buffer VAR Word(5) for SERIN functionality.
' It can still be accessed as buffer(0), buffer(1), etc.  However,
' buffer0, buffer1, etc. should be used in SERIN commands with variations
' of WAIT.

  buffer0       VAR    Byte         ' Buffer - Start char = $ff
  buffer1       VAR    Byte         ' Message Index value
  buffer2       VAR    Byte         ' Command
  buffer3       VAR    Byte         ' Argument 1 (return data 1)
  buffer4       VAR    Byte         ' Argument 2 (return data 2)
  buffer        VAR    buffer0      ' For standard array indexing
  msgIndex      VAR    Byte         ' message index
  rxc           VAR    Byte         ' Receive Clear

' -----[ Initialization ]--------------------------------------------------

Program_Start:
  LOW LED
  DEBUG CLS


  PAUSE 1000                        ' Wait for the RBT-001 radio to be ready.

  msgIndex = 0


  buffer0 = $FF                     ' Connection packet
  buffer1 = msgIndex
  buffer2 = $CC
  buffer3 = 100
  buffer4 = 100
  GOSUB Set_Servo_Speed


  DEBUG CR,"Waiting connection..."  ' wait for connection request
  SERIN BT_RX, BT_SPEED, [WAITSTR buffer \ 3, buffer3, buffer4]

' -----[ Program Code ]----------------------------------------------------

DO
```

```
  SELECT buffer2
  CASE $CC                       ' Connect
    HIGH LED
    msgIndex = 0
    DEBUG CR,"Connected"
    GOSUB Reply
  CASE $DD                       ' Disconnect
    LOW LED
    DEBUG CR,"Disconnected"
    GOSUB Reply
    GOTO Program_Start
  CASE $11                       ' Servo
    DEBUG CR,"Servo"
    GOSUB Set_Servo_Speed
    GOSUB Reply
  CASE ELSE                      ' Error
    buffer2 = $EE
    GOSUB Reply
  ENDSELECT

 Resume:                         ' If Message not rcvd, try again

 PULSOUT 13, tLeft              ' Servo control pulses
 PULSOUT 12, tRight

 SERIN BT_RX, BT_SPEED, 10, Resume, ' Get next command
     [WAITSTR buffer \ 2, buffer2,
     buffer3, buffer4]

 PULSOUT 13, tLeft              ' Servo control pulses again
 PULSOUT 12, tRight
LOOP

Reply:
  msgIndex = msgIndex + 1        ' Increment message index for reply.
  buffer1 = msgIndex            ' Next message from PC has to use reply's buf[1].
  SEROUT BT_TX, BT_SPEED, [STR buffer \5]


' -----[ Subroutines ]----------------------------------------------------

Set_Servo_Speed:                ' Maps to 650 to 850 with 750 stopped.
  tLeft = buffer3 + 650         ' Decode servo speed.
  tRight = buffer4 + 650
RETURN

' -----[ Data ]----------------------------------------------------------

ResetOnOff        DATA 0         ' On/off toggle w/ Reset button
RequestConnect    DATA $FF, 0, 1, 0, 0
ConnectionGranted DATA $FF, 0, 2, 0, 0
RequestCommand    DATA $FF, 0, 3, 0, 0
ServoSpeeds       DATA $FF, 0, 4, 0, 0
```