# 03_Bluetooth connectivity

Introduction

The AURIX™ TC29xB does not have native Bluetooth capabilities hence the use of an external Bluetooth module: the HC-05. This module has been chosen for is easy-to-use qualities as it provides fully qualified Bluetooth V2.0+EDR (Enhanced Data Rate) in a totally transparent way. This module and the AURIX™ TC29xB communicate through an LIN (Local Interconnect Network). The modules needs 2 wires for bidirectional communication and 2 wires for the power (see Hardware section down below)

Hardware

The module used is the HC-05 on the FC-114 board link to datasheet
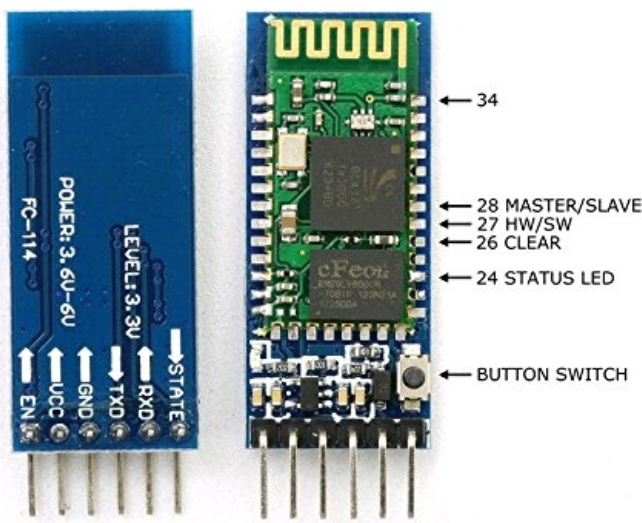
Connection HC-05->AURIX™ TC29xB

- TX pin -> 15.3 (Configurable in serial.h)
- RX pin -> 15.2 (Configurable in serial.h)
- GND -> GND
- +5V -> V_UC(+5V)

Configuration of the bluetooth module

The bluetooth module is to be configure with the values chosen for the communication. To configure the module, you will need a serial terminal on a PC and a serial-to-USB converter because the HC-05 is configured through its serial port. Once the module is connected with the computer, you can use AT commands to configure the module, you can find of these commands here. You can find a more detailed tutorial on how to configure the HC-05 here. There, the arduino is used as a serial-to-USB converter.

Software

The goal of adding the Bluetooth connectivity to the robot is for the robot to be able to be wirelessly controlled by an operator through a Bluetooth terminal (computer, smartphone, etc). The most ergonomic way for the Bluetooth connection to work is to use interrupts on the AURIX™ TC29xB which detects if something is received through Bluetooth and then interprets it as a command.

Setup of the communication

1. The setup of the code for the Bluetooth begins with the configuration of the serial communication, in order to do so, you just have to call the function serial_config and it will create a serial communication with the characteristics described in said function. First it will create an object of type ASCLIN configuration (IfxAsclin_Asc_Config), then it configures the different characteristics of the

communication:
- ○ Baudrate

* ISR priorities and interrupt target * FIFOs * pins

2. Programm the interrupts functions that are executed at receiving and transmission of data
3. Select the interrupts functions as interrupts functions example: for the function serialTxISR() $\text{IFX\_INTERRUPT}(serialTxISR,\ 0,$ $\text{ISR\_PRIORITY\_SERIAL\_TX})$; the arguments for the function IFX_INTERRUPT are (name function, vector table, priority)

Command terminal

You can communicate with the AURIX™ board with any Bluetooth capable devices, in the following I will give 2 example, one with an Android smartphone and with a Windows PC. We use both a PC and an android interface for debug purposes.

Use an Android smartphone as command terminal

There are plenty of applications that can be found on Android to communicate with another Bluetooth device,"BlueTerm 2" and "Bluetooth Terminal HC-05" are applications that have been tested for our use.

the picture above shows the interface of BlueTerm

the picture above shows the interface of Bluetooth Terminal HC-05

For these to application, you just have to turn on the bluetooth on your smartphone and then just pair it to the HC-05 module. The difference between Bluetooth Terminal HC-05 and BlueTerm is that when using the first one, you have to configure buttons were as the second one sends characters one after the other.

To create a more user friendly interface easily on an android smartphone you can use the software "MIT app inventor".

Use Windows PC as command terminal

If you decide to use your PC, two main case can arise, your PC has a Bluetooth chip in it or it does not (in which case you can use a Bluetooth dongle), just look in the device manager of windows on which COM port it is and use a serial terminal such as Putty. Keep in mind that the baudrate you are configuring in Putty has to be the same as the baudrate of the AURIX™ serial communication.

Our PC graphical interface as been developed in python for portability purposes (see wiki section GUI of the robot).

List of commands:

- 'f'-> Go forward
- 'b'-> Go Backward
- 'l'-> Turn Left
- 'r'-> Turn right
- 's'-> Stop
- 't'-> Turn 90° left
- 'u'-> Turn 90° right
- 'v'-> Turn 180° left
- 'w'-> Turn 180° right
- 'x'-> Turn 360° left
- 'y'-> Turn 360° right

Of course you can add, remove and change every of the aforementioned commands in "serial.c"