How to use iRobot Create

by Tien-Pei Chou

iRobot Create is a programmable device that was designed by iRobot at 2007. With the high programmability, the ideal of Create is for educational and experiment purpose. It is easy to implement yet powerful to execute complex commands.

The way to control the iRobot to execute the expected behavior is by sending and receiving the relative commends. One commend may compose of a singe 8-bit (one byte) signal or multiple 8-bit signals [1]. By integrating different functions, Create is capable of conducting complex actions. User can construct classes of function in console and run in the MicroBlaze. The MicroBlaze can send commends to Create and receive data from Create through the RS232 connected to the MiniDin port on Create by UART Lite/ UART Ns550 bus interface.

The user guide of iRobot Create is in [1]. The default baud rate of Create is set to 57600. Since the baud rate of UART Lite is fixed after exporting the bitstream file (unlike the UART Ns550 that can change by set baud rate command), the UART Lite has to be setup to 57600 when adding to the IP. Besides, for initialize Create, it is like an awakening signal to wake up Create from "off mode" and enter the "full mode" for fully accessing all the commands [1]. Afterwards, for controlling Create, the MicroBlaze can send the commands to Create. Each command is a serial sequence data composed of two parts: Opcode and Data Byte(s).

- **Opcode:** The first data sent to Create, telling Create which function is going to execute.

- **Data Byte(s):** Coming after the Opcode, sending the parameters (velocity, color, melody etc.) required for the function.

All the Opcode and data byte(s) are in hex. Following are four simple examples as an instruction of how to control Create. The RS232 base adders can be found in *xparater.h* in the included file in SDK.

- **Start:**
  - Opcode: 128 (decimal)
  - Data Bytes: N/A
  - Serial Sequence: [Opcode]

  The start command is the awakening signal that should always be sent once powering up Create. Or Create can not execute any commands sent from the MicroBlaze.

  XUartLite_SendByte(*RS232_Base_address,* 0x80);

- **Full Mode:**
  - Opcode: 132 (decimal)
  - Data Bytes: N/A
  - Serial Sequence: [Opcode]

The MicroBlaze has to fully access Create by making Create into the full mode. In full mode, all commands in [1] is available for implementation.

XUartLite_SendByte(*RS232_Base_address,* 0x84);

- **Drive:**
  - Opcode: 137 (decimal)
  - Data Bytes: The 1st byte and the 2nd byte are the velocity ranging from -500 mm/s ~ 500 mm/s (decimal). The 3rd and the 4th bytes are the radius ranging from -2000 mm ~ 2000 mm (decimal).
  - Serial Sequence: [Opcode][velocity][velocity][radius][radius]

  If user wants Create to drive in -200 mm/s (FF38 in hex) and 500 mm (01F4 in hex) in radius, the MicroBlaze should send a serial sequence **[89][FF][38][01][F4]** in hex by using the UART Lite sending command. The C code should be like the form below.

  XUartLite_SendByte(*RS232_Base_address,* 0x89);
  XUartLite_SendByte(*RS232_Base_address,* 0xFF);
  XUartLite_SendByte(*RS232_Base_address,* 0x38);
  XUartLite_SendByte(*RS232_Base_address,* 0x01);
  XUartLite_SendByte(*RS232_Base_address,* 0xF4);

- **Sensor:**
  - Opcode: 142 (decimal)
  - Data Bytes: Packet ID (listed in p.24 in [1])
  - Serial Sequence: [Opcode][Packet ID]

  Sensor function is more complex. Packet ID is the ID of different part of Create. By sending the ID to Create, Create will send back some parameter to the MicroBlaze (ex: bumping, current velocity, current driving distance etc.). As a result, the MicroBlaze has to receive the data from Create after sending Opcode and Packet ID by using UART Lite receiving command. The C code should be like the form below.

  XUartLite_SendByte(*RS232_Base_address,* 0x8E);
  XUartLite_SendByte(*RS232_Base_address, Packet ID_in_hex*);
  XUartLite_RecByte(*RS232_Base_address,* 0x38);

For more detail and the rest of the commands, all the resource is in [1] and should be simple to be applied with the FPGA.

Reference:

[1] iRobot Create Open Interface (enclosed in the same upload location)