**Autonomous Tractor Safety System**

**Objective 7.0** To demonstrate UART in ATSS PCB board.

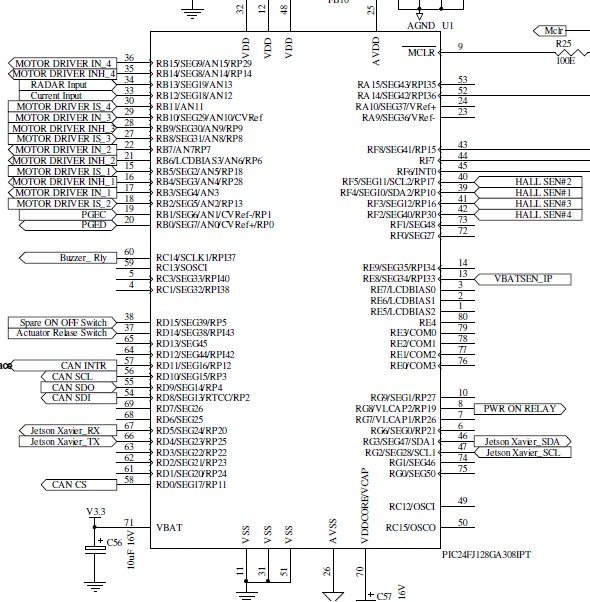
Baud rate = Fcy / (4 \* (U1BRG + 1)), Instruction cycle frequency, Fcy = Fosc/2, Fosc = 32 M, BRGH = 1, HiSpeed.

Desired baud rate = 115,200 = 32 M / (8 \* U1BRG + 1)), U1BRG = 33.7(34 approx.)

Calaculated baud rate = 32 M / (8\* (34 + 1)) = 114,285

Error % = (Calculated baud rate - Desired baud rate) \* 100 / Calculated baud rate =

(115,200 - 114,285) \* 100 / 115,200 = 0.7 %

****

**TRISF** DATA DIRECTION REGISTER F: 0x0003

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TRISF15 | TRISF14 | TRISF13 | TRISF12 | TRISF11 | TRISF10 | TRISF9 | TRISF8 |
| I/O | I/O | I/O | I/O | I/O | I/O | I/O | I/O |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TRISF7 | TRISF6 | TRISF5 | TRISF4 | TRISF3 | TRISF2 | TRISF1 | TRISF0 |
| I/O | I/O | I/O | I/O | I/O | I/O | I/O | I/O |
|  |  |  |  |  |  | 1 | 1 |
|  |  |  |  |  |  | Input | Input |

**TRISG** DATA DIRECTION REGISTER G: 0x0000

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TRISG15 | TRISG14 | TRISG13 | TRISG12 | TRISG11 | TRISG10 | TRISG9 | TRISG8 |
| I/O | I/O | I/O | I/O | I/O | I/O | I/O | I/O |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| TRISG7 | TRISG6 | TRISG5 | TRISG4 | TRISG3 | TRISG2 | TRISG1 | TRISG0 |
| I/O | I/O | I/O | I/O | I/O | I/O | I/O | I/O |
|  | 0 | 0 |  |  |  |  |  |
|  | Output | Output |  |  |  |  |  |

**ANSG** DATA DIRECTION REGISTER G: 0x0000

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANSG15 | ANSG14 | ANSG13 | ANSG12 | ANSG11 | ANSG10 | ANSG9 | ANSG8 |
| I/O | I/O | I/O | I/O | I/O | I/O | I/O | I/O |
|  |  |  |  |  |  |  |  |
|  |  |  |  |  |  |  |  |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| ANSG7 | ANSG6 | ANSG5 | ANSG4 | ANSG3 | ANSG2 | ANSG1 | ANSG0 |
| I/O | I/O | I/O | I/O | I/O | I/O | I/O | I/O |
|  | 0 | 0 |  |  |  |  |  |
|  | DIGITAL | DIGITAL |  |  |  |  |  |

**RPINR18** PERIPHERAL PIN SELECT INPUT REGISTER 18: 0x2826

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| - | - | U1CTSR5 | U1CTSR4 | U1CTSR3 | U1CTSR2 | U1CTSR1 | U1CTSR0 |
|  |  | ASSIGN UART1 CTS TO PIN5 RPI40 | | | | | |
|  |  | 1 | 0 | 1 | 0 | 0 | 0 |
|  |  | RC3 IS U1CTS | | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| - | - | U1RXSR5 | U1RXSR4 | U1RXSR3 | U1RXSR2 | U1RXSR1 | U1RXSR0 |
|  |  | ASSIGN UART1 Rx TO PIN4 RPI38 | | | | | |
|  |  | 1 | 0 | 0 | 1 | 1 | 0 |
|  |  | RC1 IS U1RX | | | | | |

**RPOR10** PERIPHERAL PIN SELECT OUTPUT REGISTER 10: 0x0400

UART1 OUTPUT (Function 4) is U1RTS

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| - | - | RP21R5 | RP21R4 | RP21R3 | RP21R2 | RP21R1 | RP21R0 |
|  |  | RP21 OUTPUT PIN MAPPING BITS | | | | | |
|  |  | 0 | 0 | 0 | 1 | 0 | 0 |
|  |  | PIN6 RG6 RP21 IS OUTPUT FUNCTION 4 – U1RTS | | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| - | - | RP20R5 | RP20R4 | RP20R3 | RP20R2 | RP20R1 | RP20R0 |
|  |  | RP20 OUTPUT PIN MAPPING BITS | | | | | |
|  |  | 0 | 0 | 0 | 0 | 1 | 0 |
|  |  |  | | | | | |

**RPOR13** PERIPHERAL PIN SELECT OUTPUT REGISTER 13: 0x0003

UART1 OUTPUT (Function 3) is TX

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| - | - | RP27R5 | RP271R4 | RP27R3 | RP27R2 | RP27R1 | RP27R0 |
|  |  | RP27 OUTPUT PIN MAPPING BITS | | | | | |
|  |  |  |  |  |  |  |  |
|  |  |  | | | | | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| - | - | RP26R5 | RP26R4 | RP26R3 | RP26R2 | RP26R1 | RP26R0 |
|  |  | RP26 OUTPUT PIN MAPPING BITS | | | | | |
|  |  | 0 | 0 | 0 | 1 | 1 | 0 |
|  |  | PIN 6 RG6 RP26 ISS UART1 TX | | | | | |

**U1MODE** UART1 MODE REGISTER: 0x8008

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| UARTEN | - | USIDL | IREN | RTSMD | - | UEN1 | UEN0 |
| UART EN |  | UART IN IDLE MODE | Ir EN/DEC EN | U1RTS MODE |  | UART EN | |
| 1 |  | 0 | 0 | 1 |  | 0 | 0 |
| ENABLED |  | CONTINUE IN IDLE | DISABLED | SIMPLEX MODE |  | U1RX, U1TX ARE EN, U1CTS, U1RTS USED BY LATCHES | |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| WAKE | LPBACK | ABAUD | RXINV | BRGH | PDSEL2 | PDSEL1 | PDSEL0 |
| WAKE UP ON START IN IDLE | LOOPBACK MODE | AUTO BAUD EN | RX POLARITY INV | HI BAUD EN | PARITY & DATA SELECT | STOP BIT SELECT | |
| 1 | 0 | 0 | 0 | 1 | 0 | 0 | 0 |
| CONTINUE SAMPLING | DISABLED | DISABLED/COMPLETED | U1RX IDLE IS ‘1’ | ENABLED 4 BRG/BIT | 8BIT DATA NO PARITY | ONE STOP BIT | |

**U1STA** UARTx STATUS AND CONTROL REGISTER: 0x0400

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| UTXISEL1 | UTXINV | UTXISEL0 | - | UTXBRK | UTXEN | UTXBF | TRMT |
| TX INT MODE | IrDA POLARITY INV | TX INT MODE |  | UART1 TX BREAK | UART1 TX EN | UART1 TX BUFFER STATUS | TX TSR BUFFER STATUS |
| 0 | 0 | 0 |  | 0 | 1 | 0 | 0 |
| INT WHEN TRSFR TO TSR | U1TX IDLE 0 IS ‘1’, IDLE 1 IS ‘0’ | INT WHEN TRSFR TO TSR |  | SYNC TRMT BRK DISABLED/ COMLETED | ENABLED | UART1 TX BUFFER NOT FULL | TX TSR BUFFER NOT FULL |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| URXISEL1 | URXISEL0 | ADDEN | RIDLE | PERR | FERR | OERR | URXDA |
| UART1 RX INT MODE | | ADDR DETECT | RX IDLE | PARITY ERROR | FRAMING ERROR | RX BUFFER OVERRUN ERROR | RX BUFFER DATA AVAILABLE |
| 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| INT WHEN TRSFR FROM RSR TO RX BUFFER | | DISBLED | RX IS ACTIVE | NONE | NONE | NONE | EMPTY |

**Code Prg07.c**

/\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*

Source code file: Prg07.c

Author, version, date: PSS ver 1.0 01.11.24

Program function: UART

Simulation: PIC24FJ128GA308 MCU, MPLAB X IDE ver 6.05, XC16 ver 2.10

\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*\*/

/\*

Baud rate = Fcy / (4 \* (U1BRG + 1)), Instruction cycle frequency, Fcy = Fosc/2,

Fosc = 32 M, BRGH = 1, HiSpeed.

Desired baud rate = 115,200 = 32 M / (8 \* U1BRG + 1)), U1BRG = 33.7(34 approx.)

Calaculated baud rate = 32 M / (8\* (34 + 1)) = 114,285

Error % = (Calculated baud rate - Desired baud rate) \* 100 / Calculated baud rate =

(115,200 - 114,285) \* 100 / 115,200 = 0.7 %

Connect the serial cable to PC either directlt or through USB converter. Configure

Hyperterminal (or Terraterm, Putty, Realterm) for the same communication parameters:

115,200 baud rate, 8 bit data, no parity and select CTS/RTS hardware for flow of

control on the available COM port. Click Connect to start emulation

A prompt is sent from the MCU to PC. The character typed in the PC is received by

the MCU. The received character is sent back to the PC from the MCU as an echo

\*/

#include <xc.h>

#include <libpic30.h>

uint8\_t c**;**

void initializeU1**(**void**);**

uint8\_t getU1**(**void**);**

uint8\_t putU1**(**uint8\_t**);**

int main**()**

**{**

initializeU1**();** // Initialize the UART1 serial port

putU1**(**'>'**);** // Prompt

**while** **(**1**)**

**{**

c **=** getU1**();** // Wait for the character

putU1**(**c**);** // Echo the character

**}**

**}**

void initializeU1**(**void**)**

**{**

ANSG **=** 0x0000**;** // RG6, RG7 in digital mode

TRISG **=** 0x0000**;** // RG6, RG7 in output mode

TRISC **=** 0x000A**;** // RC3, RC1 in input mode

RPINR18 **=** 0x2826**;** // Peripheral Pin Select Input U1CTS(RC3 pin 5 RPI40)40 =

// 0x28, U1RX(RC1 pin 4 RPI38) 38 = 0x26

RPOR10 **=** 0x0400**;** // Peripheral Pin Select Output RP21 pin is U1RTS (Output

// function 4) - RG6 pin 6

RPOR13 **=** 0x0003**;** // Peripheral Pin Select Output RP26 pin is U1TX (Output

// function 3) - RG7 pin 7

U1BRG **=** 0x0022**;** // 0x22 = 34

U1MODE **=** 0x8008**;** // UARTEN(1), UEN(00)-U1Tx, U1Rx

// are enabled & used, U1CTS, U1RTS, BCLK1 are latch controlled, ABAUD(0)-disable

// measurement, BRGH(1)-HiSpeed, PDSEL(00)-8 bit data, no parity, STSEL(0)-1 stop bit

U1STA **=** 0x0400**;** // UTXEN(1)-Tx enabled,

**}**

uint8\_t getU1**(**void**)**

**{**

LATGbits**.**LATG6 **=** 0**;** // Assert RTS, request to send

**while** **(!**U1STAbits**.**URXDA**);** // Wait

LATGbits**.**LATG6 **=** 1**;**

**return** U1RXREG**;** // Read from Rx buffer

**}**

uint8\_t putU1**(**uint8\_t c**)**

**{**

**while** **(**LATFbits**.**LATF1**);** // Wait for CTS, clear to send

**while** **(**U1STAbits**.**UTXBF**);** // Wait while Tx buffer full

U1TXREG **=** c**;**

**return** c**;**

**}**