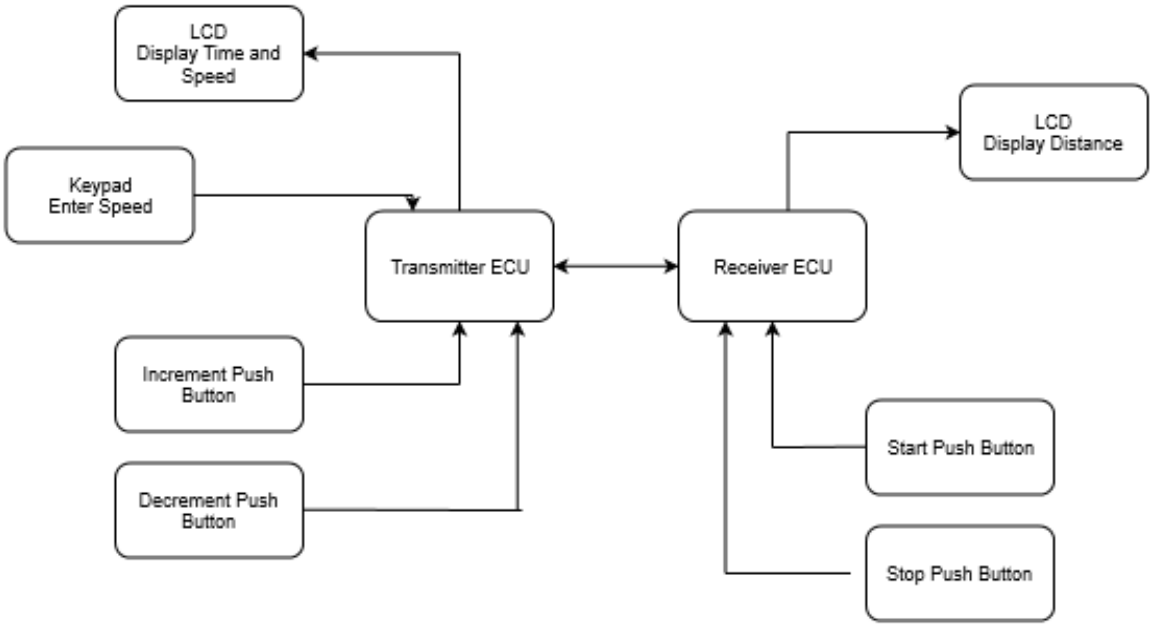
# Communication Task

**HIGH LEVEL DESIGN**

Team Dr. Strange | 17/10/2019

# Block Diagram



# Functions description

- LCD Driver

| LCD INITIALIZATION | | DESCRIPTION |
|---|---|---|
| Arguments | I/P | Void |
| | O/P | - |
| | I/O | - |
| Return | | Void |
| Prototype | | `void LCD_init(void);` |
| Description | | Initialize LCD |

| LCD DISPLAY STRING | | DESCRIPTION |
|---|---|---|
| Arguments | I/P | data<br>Type : unsigned char |
| | O/P | - |
| | I/P | - |
| Return | | Void |
| Prototype | | `void LCD_displayCharacter(uint8 data);` |
| Description | | Display required character on screen |

| LCD DISPLAY STRING | | DESCRIPTION |
|---|---|---|
| Arguments | I/P | str<br>Type : pointer to constant char |
| | O/P | - |
| | I/P | - |
| Return | | Void |
| Prototype | | `void LCD_displayString(const char *Str);` |
| Description | | Display required string on screen |

| LCD DISPLAY STRING | | DESCRIPTION |
|---|---|---|
| Arguments | I/P | Void |
| | O/P | - |
| | I/P | - |
| Return | | Void |
| Prototype | | `void LCD_clearScreen(void);` |
| Description | | Clear LCD screen |

| LCD GO TO LOCATION | | DESCRIPTION |
|---|---|---|
| Arguments | I/P | row<br>Type : unsigned char<br>col<br>Type : unsigned char |
| | O/P | - |
| | I/P | - |
| Return | | Void |
| Prototype | | `void LCD_goToRowColumn(uint8 row,uint8 col);` |
| Description | | Go to special position in LCD screen |

| LCD DISPLAY INTEGERS | | DESCRIPTION |
|---|---|---|
| Arguments | I/P | data<br>Type : signed Integer |
| | O/P | - |
| | I/P | - |
| Return | | Void |
| Prototype | | `void LCD_intgerToString(int data);` |
| Description | | Display required number on screen |

- Keypad Driver

| KEYPAD GET PRESSED | | DESCRIPTION |
|---|---|---|
| Arguments | I/P | void |
| | O/P | - |
| | I/P | - |
| Return | | unsigned char |
| Prototype | | `uint8 Keypad_getPressed(void);` |
| Description | | Get the number pressed from user |

- UART Driver

| UART INITIALIZATION | | DESCRIPTION |
|---|---|---|
| Arguments | I/P | Void |
| | O/P | - |
| | I/P | - |
| Return | | Void |
| Prototype | | `void UART_init(void);` |
| Description | | Initialize UART module |

| UART SEND | | DESCRIPTION |
|---|---|---|
| Arguments | I/P | data<br>Type : constant unsigned char |
| | O/P | - |
| | I/P | - |
| Return | | Void |
| Prototype | | `void UART_sendByte(const uint8 data);` |
| Description | | Send one byte on bus |

| UART RECEIVE | | DESCRIPTION |
|---|---|---|
| Arguments | I/P | Void |
| | O/P | - |
| | I/P | - |
| Return | | unsinged char<br>DESC.: data received |
| Prototype | | `uint8 UART_recieveByte(void);` |
| Description | | Read one byte from bus |

| UART SEND STRING | | DESCRIPTION |
|---|---|---|
| Arguments | I/P | Str<br>Type: constant unsigned char |
| | O/P | - |
| | I/P | - |
| Return | | Void |
| Prototype | | `void UART_sendString(const uint8 *Str);` |
| Description | | Send string on bus |

| UART RECEIVE STRING | | DESCRIPTION |
|---|---|---|
| Arguments | I/P | - |
| | O/P | Str<br>Type: pointer to unsigned char |
| | I/P | - |
| Return | | unsinged char<br>DESC.: data received |
| Prototype | | `void UART_receiveString(uint8 *Str);` |
| Description | | Read string from bus |

❖ Switch Driver

| SWITCH INITIALIZATION | | DESCRIPTION |
|---|---|---|
| Arguments | I/P | Port,Pin<br>Type : unsigned char |
| | O/P | - |
| | I/P | - |
| Return | | Void |
| Prototype | | `void switch_ini(uint8 Port,uint8 Pin);` |
| Description | | Initialize switch pin |

| SWITCH STATUS | | DESCRIPTION |
|---|---|---|
| Arguments | I/P | Port,Pin<br>Type : unsigned char |
| | O/P | - |
| | I/P | - |
| Return | | unsinged char<br>DESC.: Either 1 or 0 |
| Prototype | | `uint8 switch_status(uint8 Port,uint8 Pin);` |
| Description | | Check if switch is pressed or not |

❖ Timer Driver

| TIMER INITIALIZATION | | DESCRIPTION |
|---|---|---|
| Arguments | I/P | timerCfg<br>Type : pointer to struct |
| | O/P | - |
| | I/P | - |
| Return | | Func_status<br>DESC.: status of function either success or not |
| Prototype | | `Func_status TIMER_init(TIMER_cnfg_t* timerCfg);` |
| Description | | Initialize Timer module |

| TIMER STOP | | DESCRIPTION |
|---|---|---|
| Arguments | I/P | timer<br>Type : unsigned char |
| | O/P | - |
| | I/P | - |
| Return | | Func_status<br>DESC.: status of function either success or not |
| Prototype | | `Func_status TIMER_stop(uint8 timer);` |
| Description | | Initialize Timer module |

- Estimator ECU Driver

| DISPLAY TIME | | DESCRIPTION |
|---|---|---|
| Arguments | I/P | Void |
| | O/P | - |
| | I/P | - |
| Return | | Void |
| Prototype | | `void displayTime(void);` |
| Description | | Display time on LCD in terms of hours minutes and seconds |

- Measurement ECU Driver

| CALCULATIONS | | DESCRIPTION |
|---|---|---|
| Arguments | I/P | speed<br>Type: unsigned char |
| | O/P | - |
| | I/P | - |
| Return | | unsigned long long<br>DESC.: distance value |
| Prototype | | `uint64 calculations(uint8 Speed);` |
| Description | | Calculate distance in meter using speed and time in seconds |

| DISPLAY DISTANCE | | DESCRIPTION |
|---|---|---|
| Arguments | I/P | Distance<br>Type: unsigned long long |
| | O/P | - |
| | I/P | - |
| Return | | Void |
| Prototype | | `void display_distance(uint64 Distance);` |
| Description | | Display distance on LCD |

| SCREE INITIALIZATION | | DESCRIPTION |
|---|---|---|
| Arguments | I/P | Void |
| | O/P | - |
| | I/P | - |
| Return | | Void |
| Prototype | | `void screen_init(void);` |
| Description | | Initialize LCD screen with required display string |

| MEASUREMENT ECU INITIALIZTION | | DESCRIPTION |
|---|---|---|
| Arguments | I/P | Void |
| | O/P | - |
| | I/P | - |
| Return | | Void |
| Prototype | | `void Measurement_ECU_init(void);` |
| Description | | Initialize LCD and UART drivers |