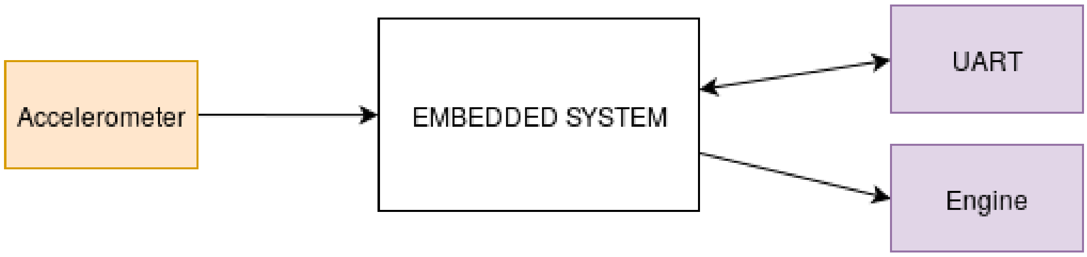
443 Final Project Report

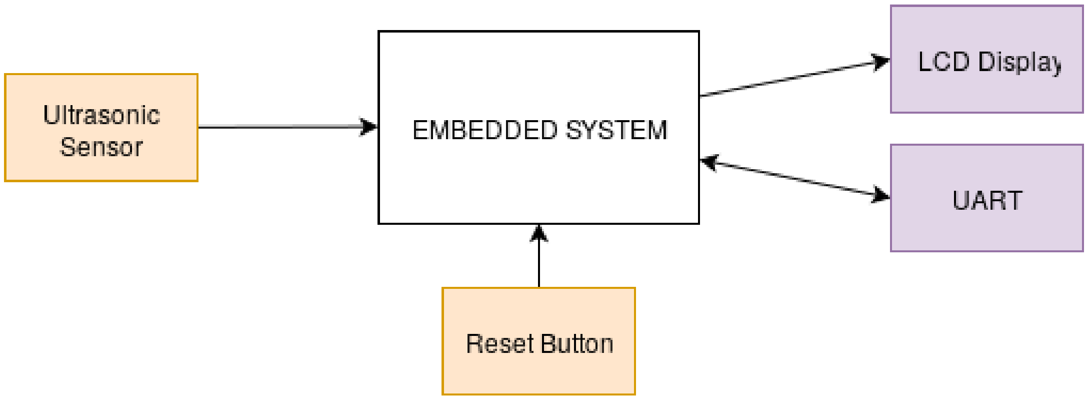
ADAMLAR

1-System Level Structural Diagram

Module: Lap Counter



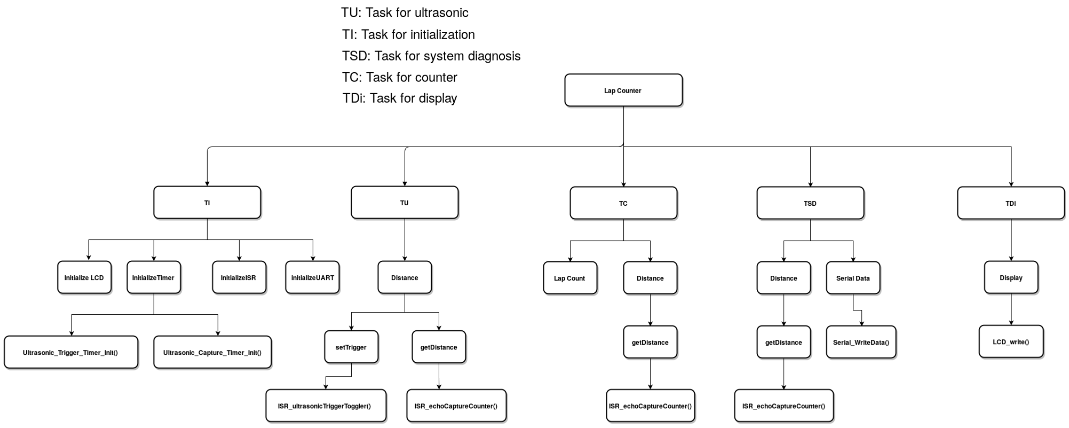
Module: Getting Speed Data from Input Device



2-

Task Decomposition Graph

Module: Lap Counter


TU: Task for ultrasonic

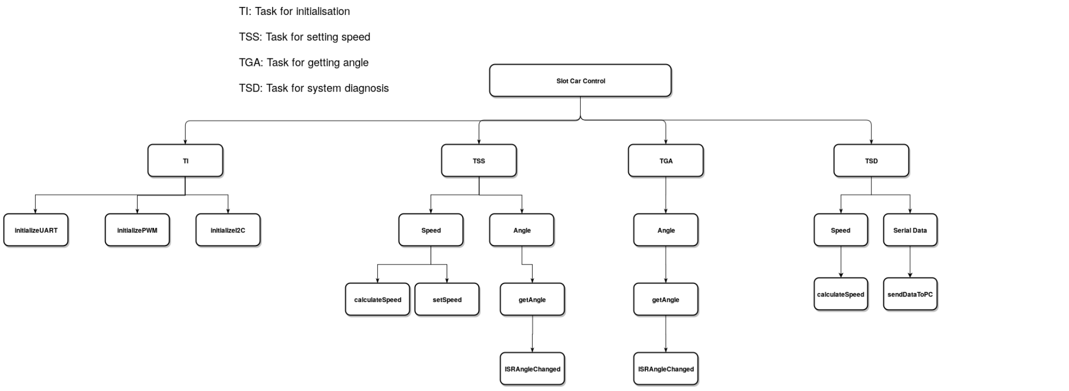
TI: Task for initialization

TSD: Task for system diagnosis

TC: Task for counter

TDi: Task for display

Module: Getting Speed Data from Input Device


TI: Task for initialisation

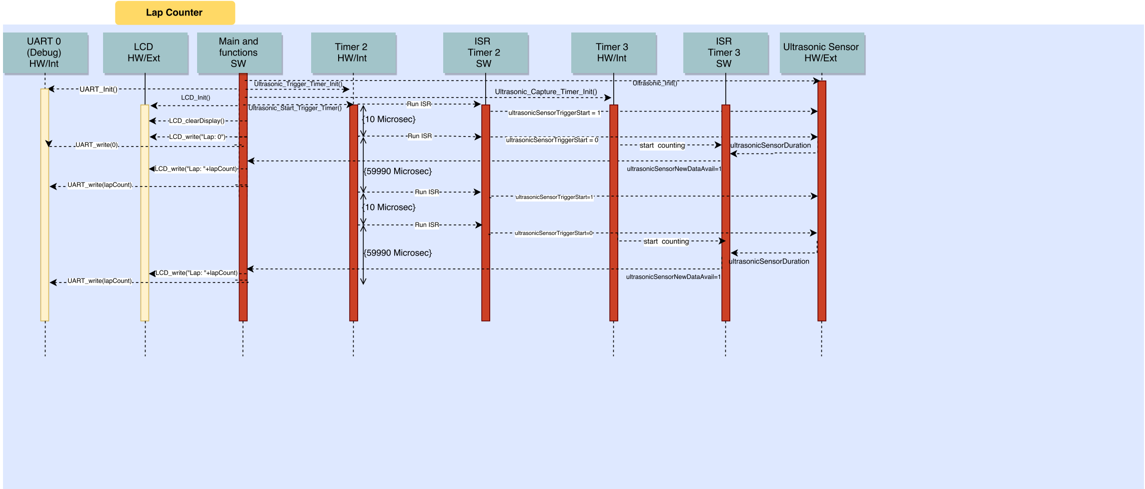
TSS: Task for setting speed

TGA: Task for getting angle

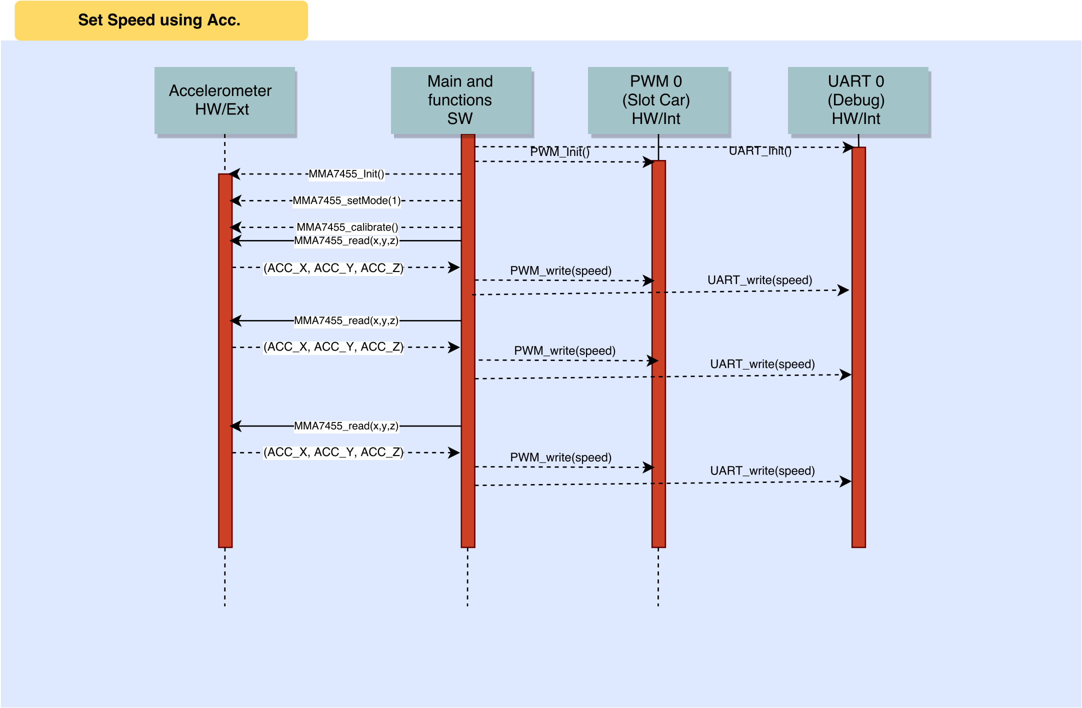
TSD: Task for system diagnosis

3-Sequence Diagram

Module: Lap Counter



Module: Getting Speed Data from Input Device



4-Coding

Module: Lap Counter

Function Name	**Function Definition**	**Objective**	**WCET** (simulation)
Ultrasonic_Init()	Initialization of the Ultrasonic sensor. Functions of the trigger and echo pins connected to the LPC board are defined in the IOCON registers	There are 2 data pins to communicate with the ultrasonic sensor. Enabling the corresponding pins in the main board, their directions' (input or output) is set for echo and trigger pins. So that, the board is able to trigger a calculation of the distance and get the incoming signals with echo pin.	18 usec
Ultrasonic_Trigger_Timer_Init()	Trigger timer initializer. The output for the trigger pin is initialized in this function	Timer 2 is used to synchronize the value(either HIGH or LOW) to the Ultrasonic sensor. To initialized it, this function is used. First the power is enabled to this section with PCONP Register , then counters (Timer counters and Prescale counters) and match values are set appropriately. Last, the function on a match("toggle" in this case) is set.	31 usec
LCD_init()	Initialization of the LCD and its pins.	LCD is initialized by setting its pins' functionalities on the board, and sending correct values to initialize the external driver, (like 0x03, 0x03,0x03,0x02)	5.18 sec

Ultrasonic_Capture_Timer_Init()	Echo timer initializer for the echo signal of the ultrasonic sensor.	First the using PCONP register, the timer is powered on. Then its match registers and timer and prescale timer registers are given the appropriate functionalities. This timer is used to count the elapsed time from the transmission of the signals to the echo of the sent signals. So that, the distance of the object could be detected.	21.3 usec
Serial_init()	Initialization of the UART 0	Using UART(Serial communication), the debug information of distance of the detected object and the lap count will be sent to the computer.	57.8 usec
TU()	Task for Ultrasonic, calculates the distance	The distance is calculated detected by ultrasonic sensor. This distance is used to detect whether the car is in the range or not.	30.6 microsec
TC()	Task for the lap counter	After getting the distance detected, this function checks whether the distance is smaller than the threshold, if it is, then the lap count is incremented by one.	23.0 microsec
TSD()	Task for system diagnosis, includes sending distance detected and the lap count data from UART to PC	In order to debug the code, we are using this function which will be printing the value of distance measured and the lap counted.	192 millisec
TDi()	This function sets the cursor to the appropriate position, then clears the display starting from the cursor, then writes the lap count into the LCD screen.	To meet the requirements specified, this function is used. It basically, writes the lap count into the LCD screen.	12.2 microsec

Module: Getting Speed Data from Input Device

Function Name	**Function Definition**	**Objective**	**WCET** (simulation)
TI()	Function initializes all the devices/modules used by the operating code.Basically consists of initializing accelerometer, PWM pin,	In order to use the devices in the board, we have to initialize them before using. Powering on the devices(PCONP) and giving the appropriate pin functionalities(IOCON) are done in this step.	5.86 secs
TSS()	Task for setting the speed of the car. To set a speed, first we are reading the angle of the board, from the accelerometer. Using the angle value of it, the pulse width of the output signal is modulated from 0 to 999 where 500 means that 50% duty cycle.	To meet the requirements specified, we are using the accelerometer data as input to set the speed of the slot car. After getting the angle value, the speed of the car is set by using PWM.	44.6 microsec
TGA()	Task for getting the angle of the board using the accelerometer device on it. This function reads the acceleration data from the device on the LPC board and finds the angle in the X dimension of the device.	To make the speed of the slot car controllable by an input device, accelerometer device is used and according to its angle in the X direction, the speed of the car is set. This function realizes the getting the angle data.	79.7 microsec
TSD()	Task for system diagnosis. This function is used to send the speed information to the PC using UART.	To debug the code that we wrote in this section, the data produced is sent to the PC to oversee the system working.	522 millisec

5-Scheduling

Module: Lap Counter

In this part of the project, we are using Polling with interrupts. When a new data is received using the ISR_echoCaptureCounter, the flag ultrasonicSensorNewDataAvailable is set to 1, the pseudo code is as follows:

```
ISR_echoCaptureCounter(){
if new data then
ultrasonicSensorDuration=duration; ultrasonicSensorNewDataAvailable=1;
}
```

This ISR is initiated after each 60 milliseconds and called when a signal is echoed back to the sensor. This process of repeated initialization and waiting functions are done via the usage of Timer 2 and Timer 3 together.

While the ISR is not running, the main function polls the flag ultrasonicSensorNewDataAvailable. The execution of the main loop is as follows:

```
while(true){
TU();//Poll the flag, if there is new data available, then calculate the distance
TC();//If the new distance is smaller then the threshold, increment the lap Counter
TDI();//Display the lap count value in the LCD
TSD();// System diagnosis, send relevant information using UART to debug
}
```

Module: Getting Speed Data from Input Device

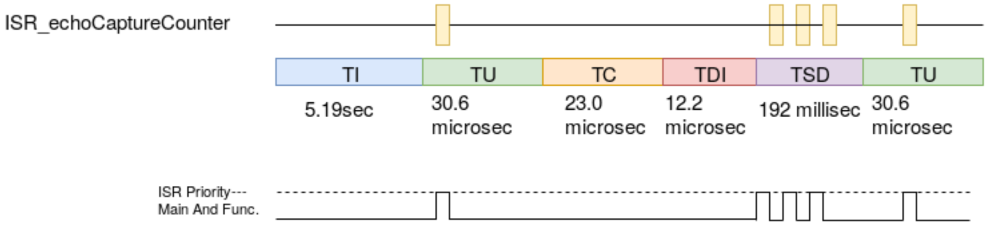
In this section, there is no interrupt used, the code executes in a cycle and reads data from the accelerometer. The communication with the accelerometer is done via I2C. An interrupt algorithm could have been ”Set a timer to repeat infinitely with a frequency, check if the value in accelerometer changed and set a flag if it is changed.”. However, this has no benefits over polling the data itself with a frequency, so we decided that it is more convenient if it is not used. The main schedule of this module program is that:

```
while(true){
TGA();//Poll the value in accelerometer and calculate angle
TSS();//Set the speed of the car using the angle value from acc.
TSD();//Send the debug information to the PC
}
```

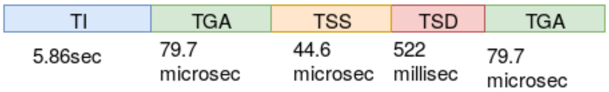
Shared Variable	Name of the function	Name of the ISR
ultrasonicNewDataAvailable	TU	ISR_echoCaptureCounter
ultrasonicSensorDuration	TU	ISR_echoCaptureCounter
ultrasonicSensorTriggerStart	TI	ISR_ultrasonicTriggerToggler

6-Timing Diagram

Module: Lap Counter



Module: Getting Speed Data From Input



Name of the ISR	Shared Variable	Priority	WCET	ACET
ISR_ultrasonicTriggerToggler	ultrasonicSensorTriggerStart	5	6.9 us	6.32 us
ISR_echoCaptureCounter	ultrasonicSensorDuration, ultrasonicSensorNewDataAvailable	1	8.5 us	6.1 us

7-Hardware Block Diagram

Component	Type ID	Quickstart Board	Base-board	Off-board
Ultrasonic	HC-SR04			x
LCD	LCM-S01601DSR			x
Potentiometre	B10K			x
Accelerometer	MMA7455		X	
LED				X
1K-Resistor				X

Expenses	Cost
120pcs 10cm male to male + male to female and female to female jumper wire	\$2.5

8-Board Pin Table

LCD PINS	LPC4088 PINS
1	GND
2	VU
3	to 2. pin of potentiometer
4 RS	P0.8 (P12)
5 RW	P0.6 (P14)
6 EN	P0.7 (P13)
11 DATA0	P0.24 (P16)
12 DATA1	P0.25 (P17)
13 DATA2	P0.26 (P18)
14 DATA3	P1.30 (P19)

Potentiometer	LPC4088 PINS
left	Vin
right	GND

Ultrasonic	LPC4088 PINS
Vcc	Vin
GND	GND
Trig Trigger	P0.9(P11)
Echo Echo	P0.23(P15)

Slot Car	LPC4088 PINS
Vcc	P1.5(P28)
GND	GND

9-Appendix

Please see the LapCountModule and GettingSpeedFromInputModule folders for the related source code.