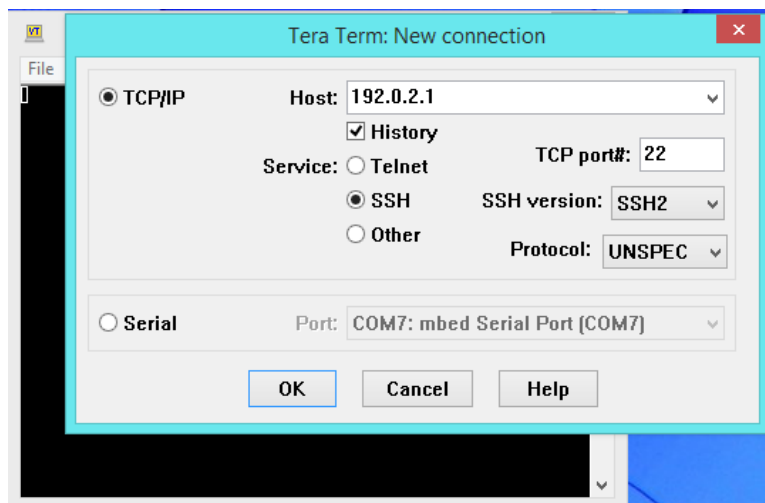
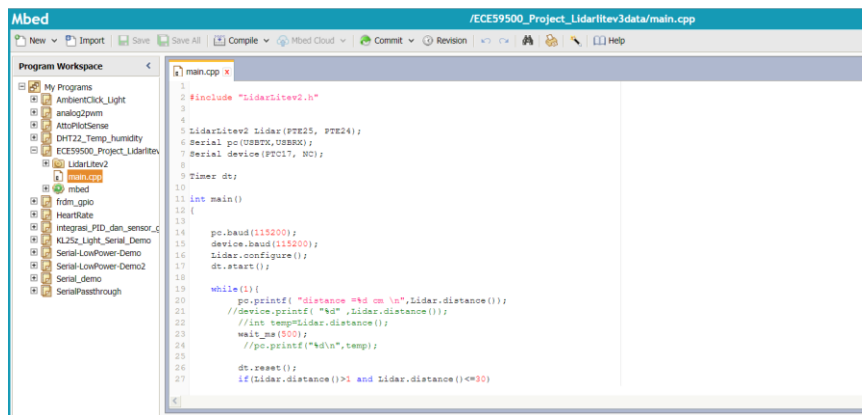


IMPLEMENTATION STEPS:

1. The Lidar Lite V3 connected to the FRDMK64F using I2C board to detect the presence of object and the distance of the object from the sensor.
2. The distance of the object calculated by the sensor is then sent to S32K144 (PTD6) using the UART interface.
3. This distance is mapped in to different rpm of BLDC motor. According to this distance, the speed of the motor will be controlled.
4. Connect the CAN High, Low and Ground pins of 2 S32K144. (Refer the connection diagram).
5. Power up the S32K144 from external supplies.
6. Connect the DEVKIT MotorGD and the Linux-BLDC motor to the slave S32K144.
7. Power up the MotorGD shield using 15V,2A power adapter.
8. Make the S32K144 EVB to work on closed loop control with hall sensors.
9. We will be using Model Based Design Toolbox for S32K Series from matlab to model our system and implement it on target MCU.

LIDATLITEV3 DATA (MBED)

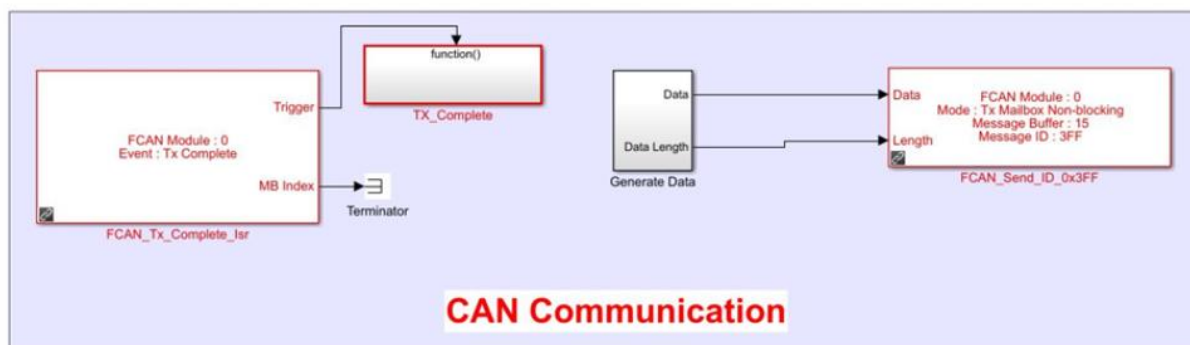
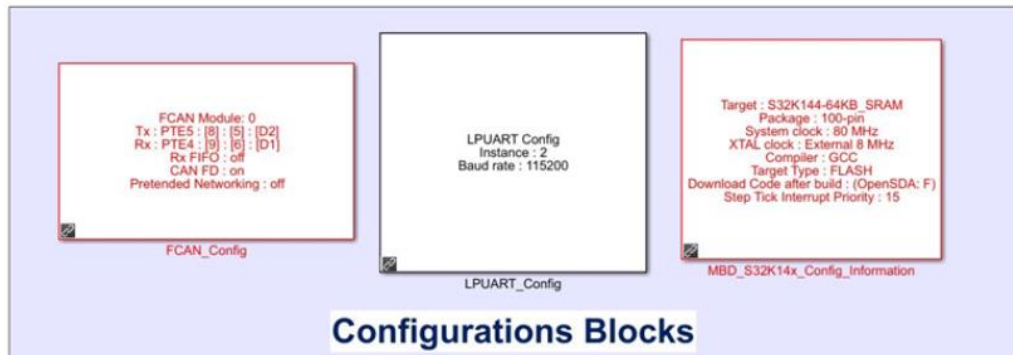
- Open mbed Compiler, compile and download the “Lidatlite” on FRDMK64F.
- Set the baud rate of Tera term to 115200 to view the Lidar sensor data.



MODEL BASED DESIGN TOOLBOX

- Execute the “ECE595_Part1_Project_Lidar_based_Motor_Control_Master” on the S32K144 which is going to act as master.
- Set the OpenSDA Drive name in PIL and Download Config to the drive name of master S32K144 and select GCC as the compiler before execution.

ECE595_Part1_Project_Lidar_based_Motor_Control_Master



- Run the “ECE595_Part1_Project_Lidar_based_Motor_Control_Slave” on the slave S32K144 with the BLDC motor.
- Set the OpenSDA Drive name in PIL and Download Config to the drive name of master S32K144 and select GCC as the compiler before execution.

[illegible]

- "Lidar Based Motor Control"**

Lidar on FIDMK14 to detect the obstruction

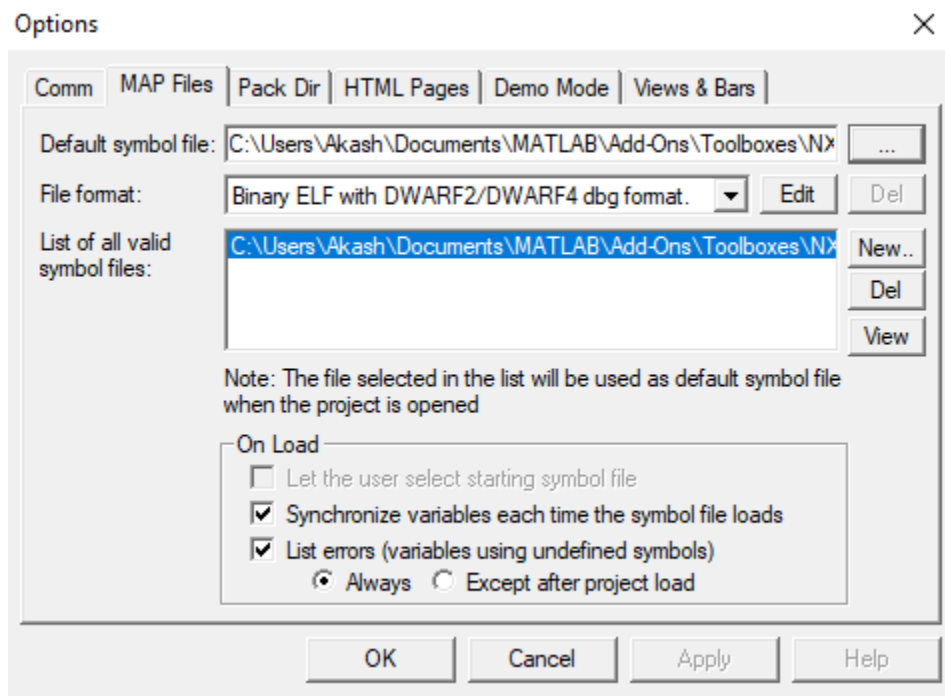
S32K144

CAN PROTOCOL

RPM control accordingly

Distance(in cm)	RPM
0-30	0
30-60	400
60-90	800
90-180	1200
180-240	2400
240-270	2800
270-300	3200
above 300	4000

- Select Project, Options then MAP Files



- Then open the elf file that is generated when building the UART protocol model in model-based design toolbox.
- Check your com port at the device manager and then Select “Comm” and set the communication.
- Then select “tools” then “Connection Wizard” to establish the communication
- To display the speed of motor, create a new watch variable. In the address, select “MotorSpeedReqFreemaster” to display the RPM of motor.
- Select a sampling period of 100ms/fastest and change the variable name to “MotorSpeedReqFreemaster.
- Create a scope to watch the RPM on Graphical panel.

Free MASTER SCOPE: