Temperature sensing is a very important part of fully functional automobile. As the automotive industry soars ahead in the field of intervehicle communication, data is readily available to process further.

There are sensors all around the vehicle as well as inside them. One of the important sensors is a temperature sensor.

The climate control unit in the car is equipped with a temperature sensor. The engine temperature is monitored using a temperature sensor.

In my project we have tried to emulate such situations and used 2 evaluation boards to transfer the temperature from the remote location to the main control unit.

The temperature sensor in use is Si7021 from Silicon Labs.

Brief Explanation:

The NXP’s Model-Based Design Toolbox gives an integrated development environment and tool chain, and generate necessary software required configure complex applications such as communication protocols CAN, SPI, I2C, UART and sensor based applications as well as motor control.  
The toolbox includes integrated [SIMULINK](https://www.mathworks.com/products/simulink.html) embedded target for NXP MCUs, peripheral device blocks and drivers, here are some important features:  
1. It can generate code for standalone application using direct download to target support  
2. Build motor control library optimized blocks including Park/Clarke transforms, digital filters, and general functions.  
3. CAN, SPI, PIT timer, Sine Wave Generation, eTimer, PWM and A/D are included in the IO blocks  
4. SIL and PIL test and other seamless integrations using embedded coder, which is easier to process.

Check the config file and verify the driver name is the same:

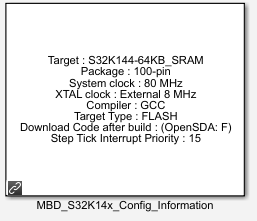


Fig: Config files

FlexIO module:

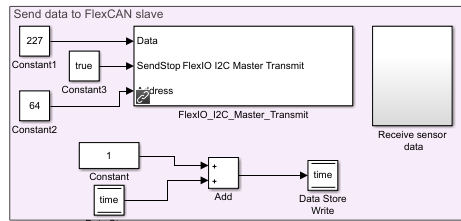
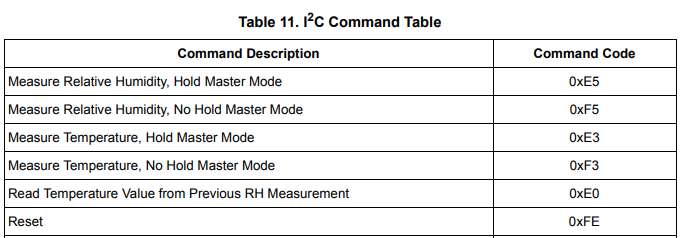


Fig: FlexIO Module

The FlexIO\_I2C\_Master\_Transmit creates an I2C packet to command the sensor to send the date to the Master.

227 is the command id which tells slave to measure the temperature and send it back

64 is the slave address.

6.4.3 The FlexIO\_I2C\_Master\_Receive module returns the raw temperature value.

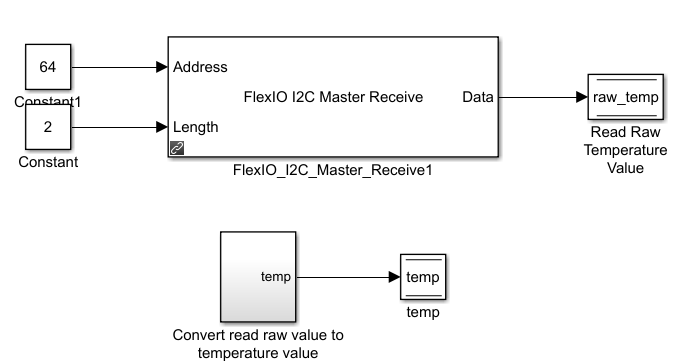


Fig: Receive raw temp and convert it to degree Celsius blocks

After this the raw temperature value is converted to a Celsius value using a sub system:

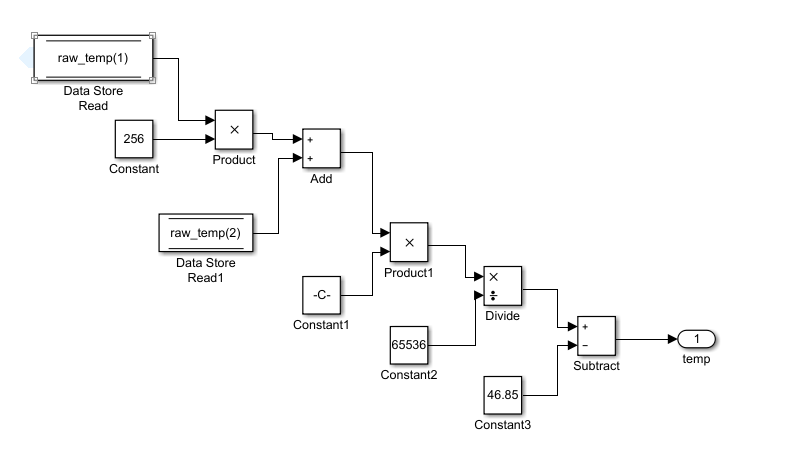
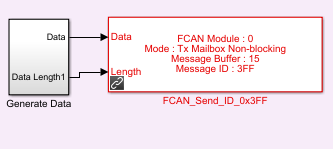


Fig: Converting raw temp to degree Celsius

This raw value is stored in a memory location and sent to the CAN module to be transmitted.



6.4.4 This module will pack the data into the CAN-FD protocol and transmit using message ID: 0x3FF.

The ISR will interrupt the routine to transmit this data:

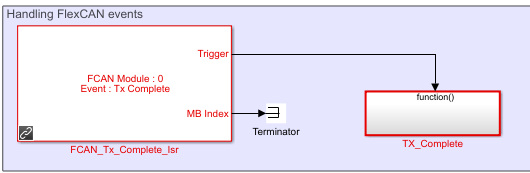
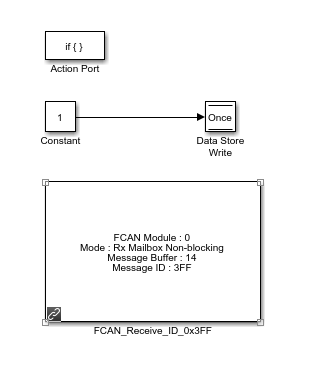
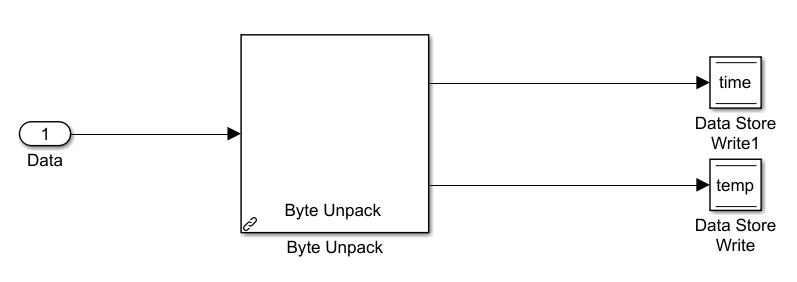


Fig: Handling FlexCAN events

6.4.5 The slave has been initialized and it is waiting for an interrupt from the master:

Once the data is received, the data is unpacked:





This unpacked data is then stored in a memory location temp.