
DC Motor Control with Touch Interface and IR Communication

Introduction

Authors: Namrata Dalvi, Ankit Phougat, Microchip Technology Inc.

Intelligent electronics have become an integral part of today's industries and smart homes. Many real-time control applications can be carried out using such electronic devices, and the communication mode used is the key to all these applications.

Infrared light, commonly referred to as "IR", is one such communication that is very common and easy to use. It consumes low power and is cost effective to implement in applications involving transmission of information and commands to control various home appliances. Some commonly used home appliances that involve IR communication are remote controls for television and air conditioners. IR communication is also used in security and safety products, like smoke detectors and occupancy sensors. There are many products that use motor control and IR communication, such as printers, toy cars, fans, etc.

The PIC18 Q10 series of microcontrollers (MCUs) comes with a wide variety of features to serve such applications. The devices include analog and digital peripherals, communication modules and a host of Core Independent Peripherals (CIPs) that work independently and reduce the CPU overhead. These devices are well suited for battery-powered applications because of their low-power consumption. In addition, they offer precise control capability due to predictable timing and monitoring.

This application note covers one such application that combines IR communication and DC motor control by using intelligent peripherals of the PIC18 Q10 microcontroller.

Features

This application note features the following:

- Highlighting Features of PIC18 Q10 MCUs for Real-Time Control Applications
- An Overview of Capacitive Touch Sensing Technique (mTouch[®] sensing solution) from Microchip
- An Overview of IR Communication and NEC Protocol
- Details of the IR Transmitter and Touch Interface Implementation Using ADC² with CVD, CLCs and DSM Peripherals
- Details of the IR Receiver and DC Motor Control Implementation Using CCP and PWM Peripherals

Table of Contents

Introduction.....	1
1. Features of PIC18 Q10 Devices.....	4
2. Capacitive Touch Sensing.....	6
2.1. Capacitive Touch Sensing Using ADC ² and Hardware Capacitive Voltage Divider (CVD).....	6
2.2. mTouch Capacitive Sensing Library Module for MPLAB X Code Configurator.....	6
3. IR Communication.....	7
3.1. IR Communication Protocols.....	7
3.2. NEC Protocol	8
4. Application Overview.....	10
5. Hardware Overview.....	12
5.1. Curiosity HPC Development Board.....	12
5.2. QT7 Xplained Pro	12
5.3. IR Click.....	13
5.4. DC Motor 8 Click.....	13
5.5. DC Motor.....	14
6. Firmware Overview	15
6.1. Software Tools.....	15
6.2. Touch Interface and IR Transmitter.....	15
6.2.1. Touch Interface.....	16
6.2.2. IR Transmitter.....	16
6.2.3. MCU Clock and Peripheral Configurations Using MCC.....	17
6.2.4. Firmware Operation.....	25
6.2.5. Application Flowchart for Touch Interface and IR Transmitter.....	27
6.2.6. Program and Data Memory Requirements	27
6.3. IR Receiver and DC Motor Control.....	27
6.3.1. IR Receiver.....	28
6.3.2. DC Motor Control.....	28
6.3.3. MCU Clock and Peripheral Configurations Using MCC.....	28
6.3.4. Firmware Operation.....	34
6.3.5. Application Flow.....	36
6.3.6. Program and Data Memory Requirements	37
7. Demo Operation.....	38
7.1. Demo Setup for the Touch Interface and IR Transmitter.....	38
7.2. Demo Setup for the IR Receiver and DC Motor Control.....	40
8. The Possible Application Use Cases.....	42
9. Conclusion.....	43

10. References.....	44
11. Appendix.....	45
11.1. Advantages and Challenges of IR Communication for Real-Time Control Applications.....	45
The Microchip Web Site.....	46
Customer Change Notification Service.....	46
Customer Support.....	46
Microchip Devices Code Protection Feature.....	46
Legal Notice.....	47
Trademarks.....	47
Quality Management System Certified by DNV.....	48
Worldwide Sales and Service.....	49

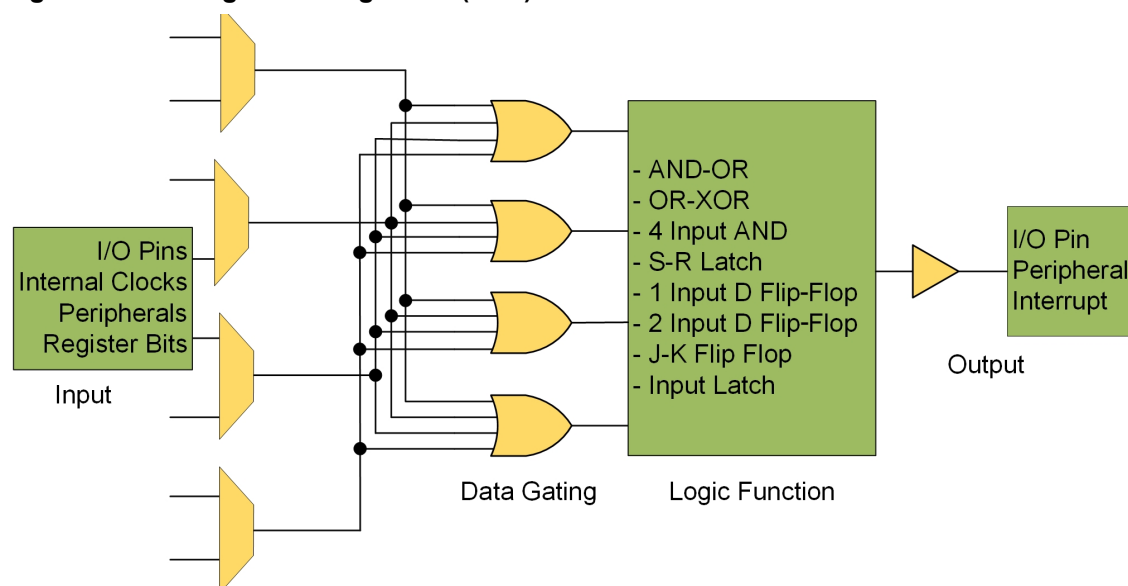
1. Features of PIC18 Q10 Devices

The PIC18 Q10 series of microcontrollers is equipped with a 10-bit ADC with Computation (ADC²) that features hardware Capacitive Voltage Divider (CVD) for capacitive touch sensing applications. These devices offer a set of Core Independent Peripherals (CIP) such as Configurable Logic Cell (CLC), Data Signal Modulator (DSM), Hardware Limit Timer (TMR2/4/6+HLT) and Peripheral Pin Select (PPS), providing increased design flexibility and lower system cost.

This section highlights the features of PIC18 Q10 MCU devices that are essential for real-time control applications:

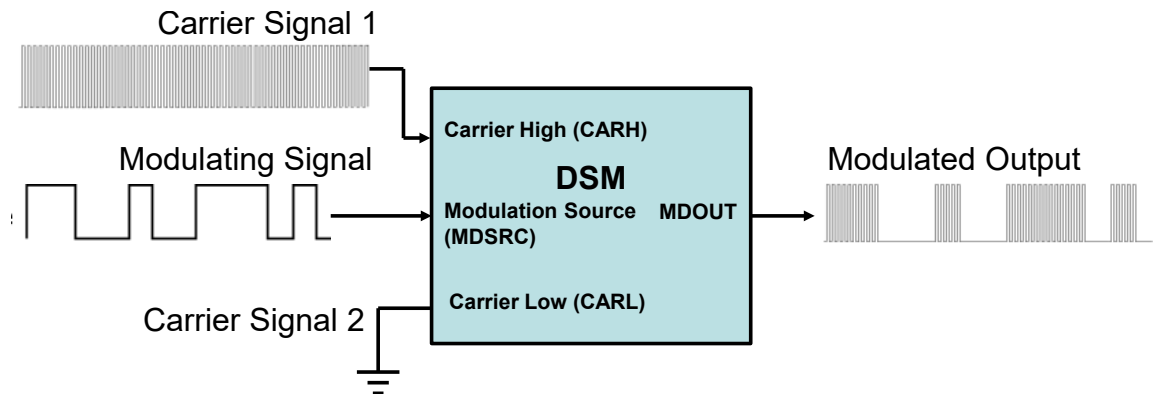
- Operating Speed Up to 64 MHz Over the Full V_{DD} Range with 62.5 ns Minimum Instruction Cycle
- Three 8-Bit Timers (TMR2/4/6) with Hardware Limit Timer (HLT)
- Four 16-Bit Timers (TMR0/1/3/5)
- Memory:
 - Up to 128 Kbytes program Flash memory
 - Up to 3615 bytes data SRAM memory
 - Up to 1024 bytes data EEPROM with programmable code protection and direct, indirect and relative addressing modes
- Different Power-Saving Modes with Peripheral Module Disable (PMD) and Extreme Low-Power Mode (XLP)
- Up to 35 I/O Pins and One Input Pin, with Interrupt-on-Change (IOC) on All Pins
- Core Independent Peripherals:
 - Eight Configurable Logic Cells (CLCs) with integrated combinational and sequential logic: Provides programmable logic that operates outside the speed limitations of software execution. CLCs can be used to build custom logic functions and to internally connect other peripherals like timers, PWMs, serial ports and I/O pins, allowing hardware customization with unprecedented ease.

Figure 1-1. Configurable Logic Cell (CLC)



- Data Signal Modulator (DSM): Provides the user with the ability to mix a data stream (modulating signal) with a carrier signal to produce a modulating output.

Figure 1-2. Data Signal Modulator (DSM)



- Hardware Limit Timer (TMR2/4/6+HLT):
 - Hardware monitoring of missed periodic events and fault detection
 - General purpose 8-bit timer/counter with external Reset capabilities
 - Different modes, for example, Rollover Pulse, Monostable, One Shot, and One Shot with Edge Trigger
- Capture/Compare/PWM (CCP) modules with two CCPs, 16-bit resolution for Capture/Compare modes and 10-bit resolution for PWM mode
- Two 10-Bit Pulse-Width Modulators (PWM)
- Peripheral Pin Select (PPS) Enables Pin Mapping of Digital I/O
- 10-Bit Analog-to-Digital Converter with Computation (ADC²):
 - 35 external channels
 - Conversion available during Sleep
 - Four internal analog channels
 - Internal and external trigger options
 - Automated math functions on input signals:
 - Averaging, filter calculations, oversampling and threshold comparison
 - 8-bit hardware acquisition timer
 - Hardware Capacitive Voltage Divider (CVD) support:
 - 8-bit precharge timer
 - Adjustable sample and hold capacitor array
 - Guard ring digital output drive

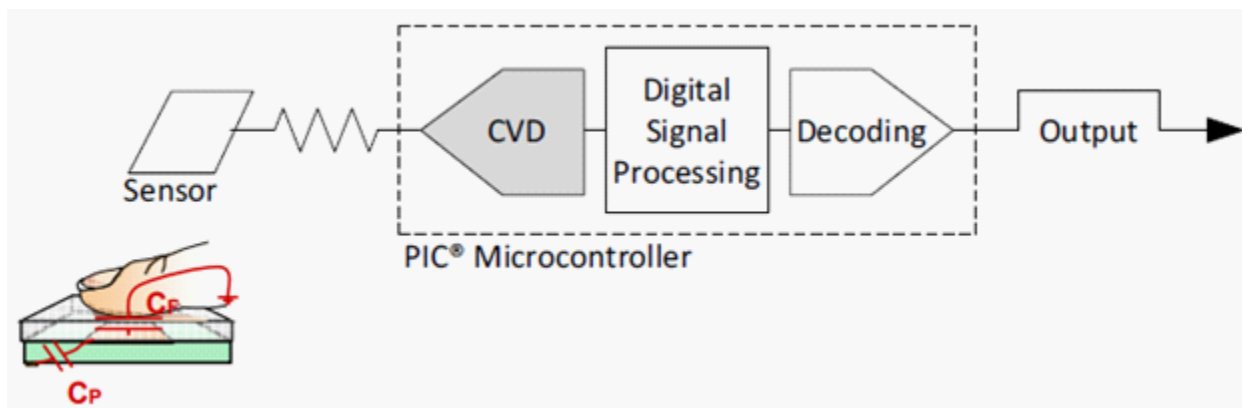
2. Capacitive Touch Sensing

Many products that require human interactions integrate capacitive touch interface as it is aesthetically appealing, robust, flexible to design and very cost effective. Microchip's mTouch Sensing Solution is a touch sensing solution based on changes in capacitance. Many PIC[®] microcontrollers have ADC with Computation (ADC²) and hardware Capacitive Voltage Divider (CVD) support, which is required for capacitive touch sensing. Microchip also provides easy-to-use mTouch Capacitive Sensing Library Module for MPLAB[®] X Code Configurator (MCC). Adding mTouch library to a firmware project enables touch sensing channels in an application, which are required for the touch sensors.

2.1 Capacitive Touch Sensing Using ADC² and Hardware Capacitive Voltage Divider (CVD)

There are several techniques for capacitive touch detection, the most common being Capacitive Voltage Division (CVD). Microchip's differential CVD acquisition technique is a charge/voltage-based technique to measure relative capacitance on a pin using only an ADC and a minimal amount of digital processing overhead to implement advanced touch sensing. The CVD sets up a square charging wave and measures the decay result, then averages a number of decay results and compares that to some threshold.

Figure 2-1. CVD system



For more details about CVD, please refer to the available application note [AN1478: mTouch[®] Sensing Solution Acquisition Methods Capacitive Voltage Divider](#) on our website.

2.2 mTouch Capacitive Sensing Library Module for MPLAB X Code Configurator

The mTouch capacitive sensing library module allows for quick and easy C code generation of the capacitive touch button, proximity sensor and slider/wheel solutions.

This library module uses MCC's Graphical User Interface (GUI) to accomplish the configuration:

- Setting multiple mTouch parameters
- Enabling various mTouch features
- Generating the necessary C code to program onto a PIC microcontroller

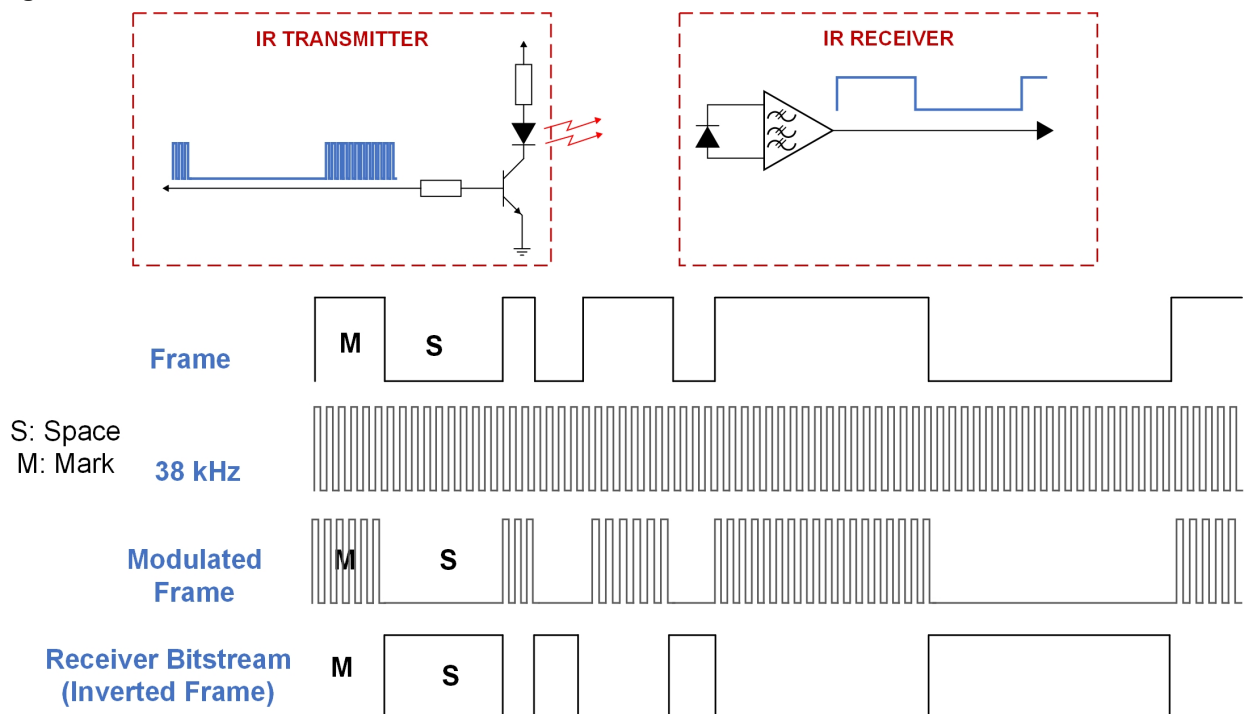
For more details about how to setup a basic mTouch project, refer to [mTouch[®] Capacitive Sensing Library Module for MPLAB[®] X Code Configurator User's Guide](#).

3. IR Communication

Infrared (IR) is a very common wireless communications technology because of its low cost and ease of use. Infrared light has a slightly longer wavelength than visible light, which is undetectable to human eyes. In everyday life, there are many sources of infrared light, such as the sun, light bulbs and any object that dissipates heat.

Hence, in real world applications there is a need to modulate the IR signal with a carrier signal. This is to avoid the possible interference with the rest of the IR signals that are being emitted by the surrounding IR sources. Usually a fixed carrier frequency, typically between 33 and 40 kHz or 50 to 60 kHz, is used for modulation. The most commonly used protocol is the NEC IR protocol, which specifies a carrier frequency of 38 kHz.

Figure 3-1. IR Transmission



As shown above, a typical IR frame is comprised of marks and spaces. Space is the default state of the signal (logic low in “Frame” and “Modulated frame” signals). Mark is when the transmitter is active and switching at modulation frequency. The duration of mark and space is where information to be conveyed is encoded.

Different protocols use marks and spaces differently. Marks and spaces are different depending on the receiver. In the IR receiver, the mark is logic low and space is logic high.

3.1 IR Communication Protocols

There are various standard IR data communication protocols in use. Sony® manufactured consumer devices of different types that share a common proprietary protocol, called S-link. The RECS-80 and RC-5 codes developed by Philips have been casually referred to as international standards. However, the RECS-80 protocol was prone to interference and was quickly replaced by the RC-5 protocol.

The rapidly expanding requirements for newer categories of electronics products, like DVD players, cable boxes, or DVRs have led Philips to replace the RC-5 protocol with the newer RC-6 protocol that has an expanded set of devices (256 versus 32) and commands per device (256 versus 64 in RC-5 and 128 in RC-5x).

The major Japanese consumer electronics manufacturers almost universally adopted a protocol developed and administered by NEC (now Renesas Electronics). In the NEC protocol, each manufacturer is assigned a unique code contained in the transmitted command, avoiding the possibility of false triggering by other remote handsets.

These transmission protocols are easily created and/or decoded with general purpose 8-bit microcontrollers. The transmission of the IR commands requires only a microcontroller and an infrared Light-Emitting Diode (LED), available from a wide variety of sources. The reception of the modulated commands for RC-5, RC-6, and the NEC protocols is easily accomplished with specialized IR receivers. In this application, the NEC protocol is being used.

3.2 NEC Protocol

The NEC IR transmission protocol uses pulse distance encoding of the message bits. Each pulse burst (mark – IR LED transmitter ON) is 562.5 μ s in length, at a carrier frequency of 38 kHz. Logical bits are transmitted as follows:

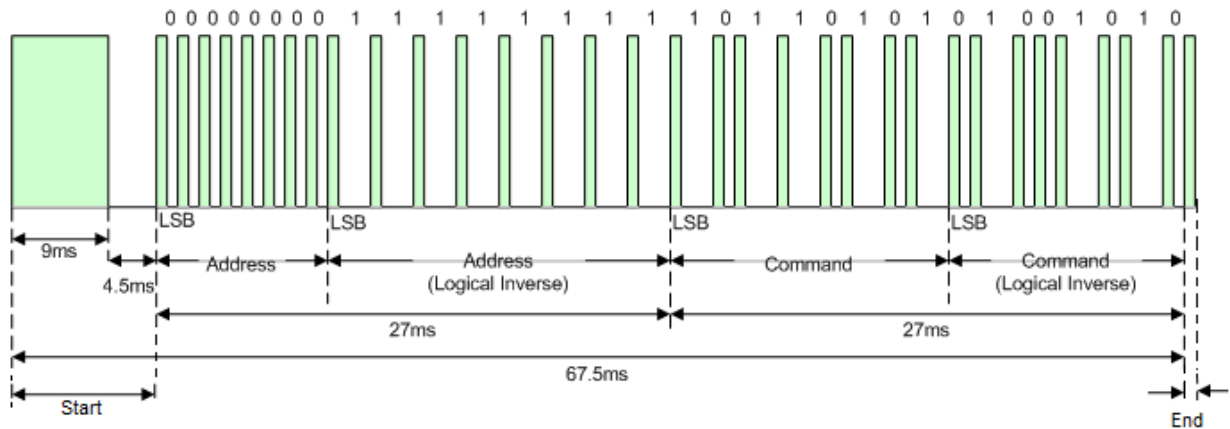
- Logical '0' - a 562.5 μ s pulse burst followed by a 562.5 μ s space; a total transmit time of 1.125 ms.
- Logical '1' - a 562.5 μ s pulse burst followed by a 1.6875 ms space; a total transmit time of 2.25 ms.

While transmitting or receiving remote control codes using the NEC IR transmission protocol, the codec performs optimally if the carrier frequency used for modulation/demodulation is set to 38.222 kHz. When a key is pressed on the remote controller, the message transmitted consists of the following, in order:

- A 9 ms leading pulse burst (16 times the pulse burst length used for a logical data bit)
- A 4.5 ms space (9 ms mark + 4.5 ms space = 13.5 ms start sequence)
- The 8-bit address of the receiving device
- The 8-bit logical inverse of the address
- The 8-bit command
- The 8-bit logical inverse of the command
- A final 562.5 μ s pulse burst to signify the end of message transmission.

The four bytes of data bits (two bytes of address and two bytes of command) are sent sequentially with the Least Significant bit (LSb) first. The following figure illustrates the format of an NEC IR transmission frame for an address 0x00 (00000000b) and a command 0xAD (10101101b).

Figure 3-2. Example Message Frame Using the NEC IR Transmission Protocol



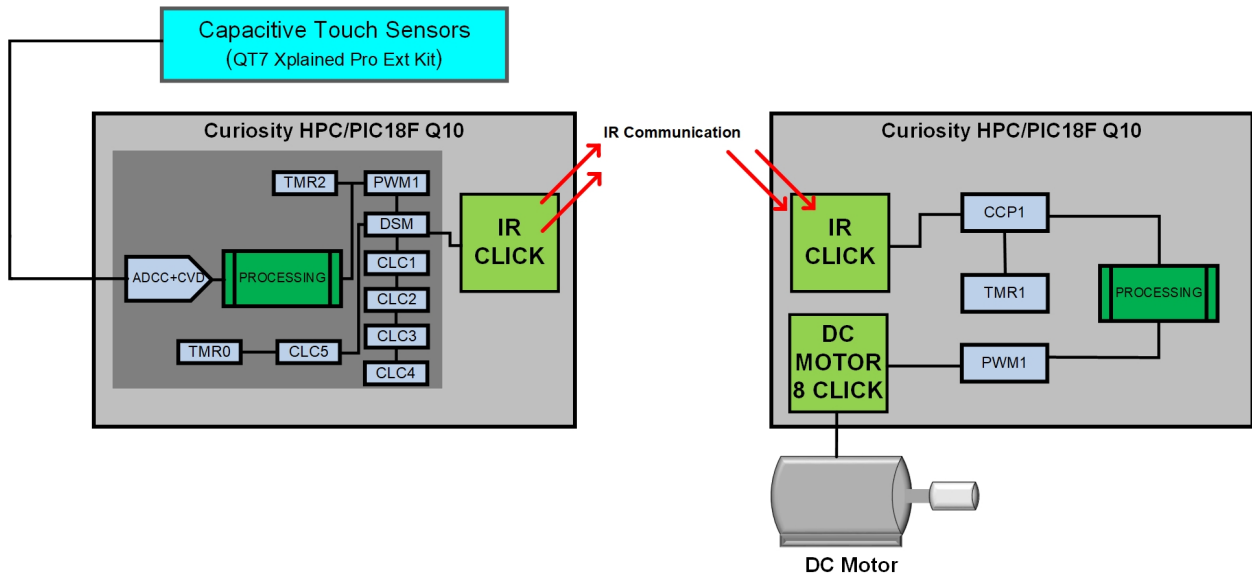
The time required to transmit one byte of command using NEC IR transmission protocol as depicted in the figure above is calculated below.

- It requires 27 ms each to transmit the 16 bits for the address (address + inverse) and the 16 bits for the command (command + inverse). This comes from each of the 16-bit blocks ultimately containing eight '0's and eight '1's – requiring $(8 * 1.125) \text{ ms} + (8 * 2.25) \text{ ms} = 9 + 18 \text{ ms} = 27 \text{ ms}$.
- It takes 68.625 ms to transmit the message frame: a 13.5 ms start sequence, 27 ms for receiver address, 27 ms for command and a final 562.5 µs pulse burst that signifies the end of message.

4. Application Overview

In this application note, a simple real-time control application is implemented using PIC18 Q10 MCUs. This application uses a combination of CIPs to build a capacitive touch sense interface, IR transmitter, IR receiver and DC motor control units.

Figure 4-1. DC Motor Control Using Touch Interface and IR Communication

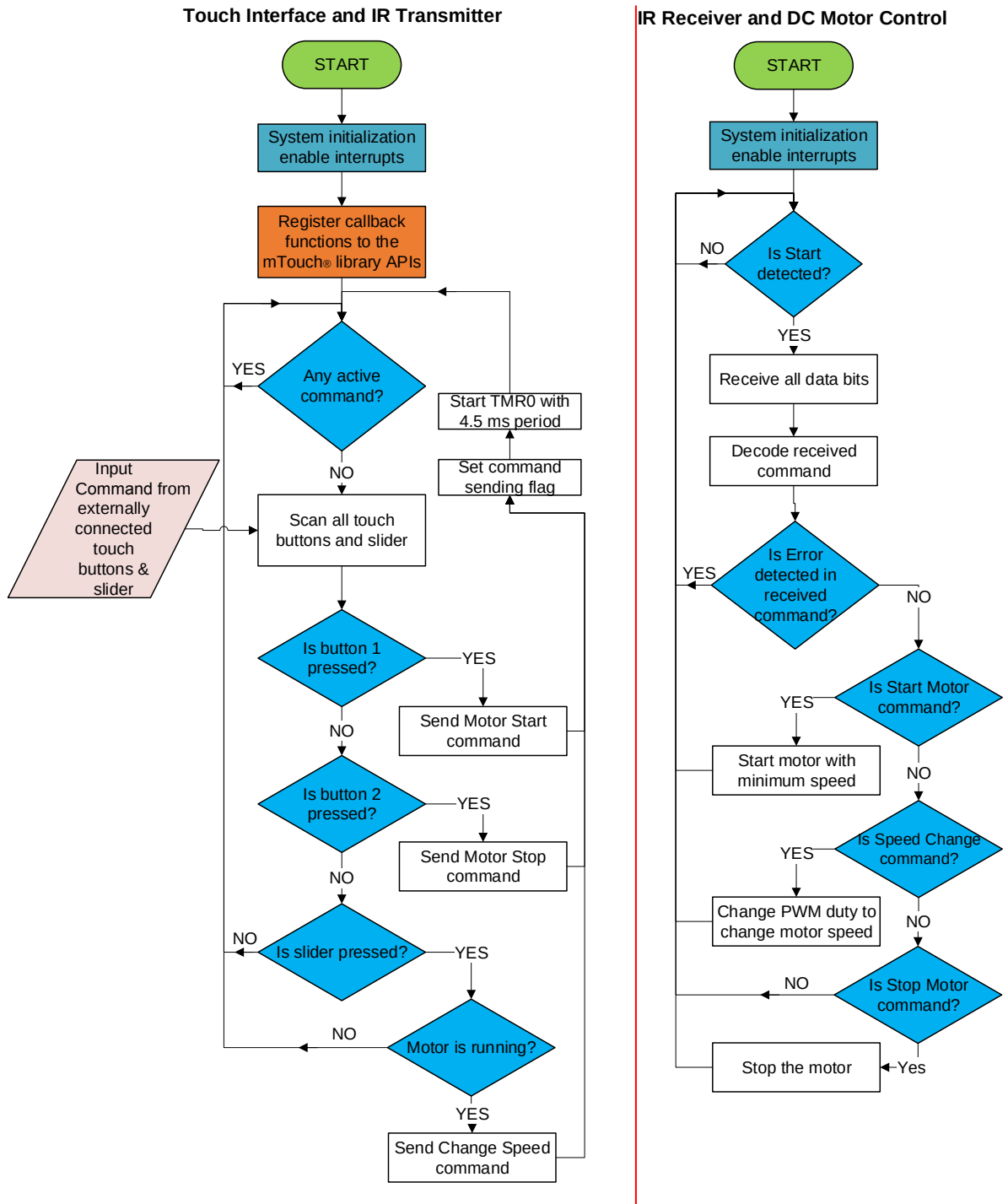


The ADC² module of the PIC microcontroller continuously acquires the capacitive touch sensors data using CVD technique and monitors the status of the touch sensors. The QT7 Xplained Pro extension board consists of two touch buttons and a slider. Button 1 of the two touch buttons on QT7 Xplained Pro extension board is used to start the motor. When the DC motor is stalled and button 1 is pressed, the command to Start the motor is transmitted. On the other hand, when the motor is running and button 2 is pressed, the command to Stop the motor is transmitted. The slider can be used to increase or decrease the speed of the DC motor. The microcontroller generates predefined commands depending upon appropriate sensor detection.

The generated commands are transmitted to the receiver over IR communication. The IR transmitter is implemented using CLCs, PWM and DSM peripherals of the PIC18 Q10 microcontroller and uses NEC protocol to encode the commands to be transmitted. The IR signal output from the DSM module is supplied to the IR LED on IR click board.

On the receiver side, the IR click board receives IR commands using the on-board IR receiver module. The received signal is fed to the CCP of PIC18 Q10 MCU on the curiosity HPC board. The microcontroller decodes the received commands using CCP module. Corresponding control action is taken to start/stop and change the speed of the DC motor using the PWM peripheral of the microcontroller and the DC motor click board.

Figure 4-2. Application Flowchart



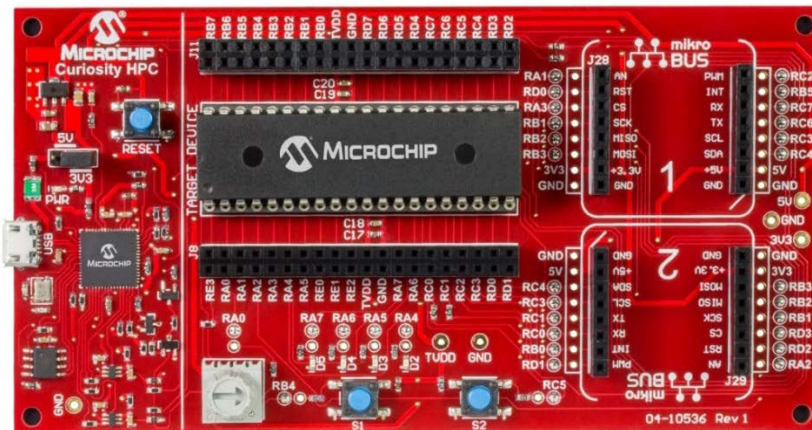
5. Hardware Overview

This application uses existing evaluation boards. The [Touch Interface and IR Transmitter](#) section is comprised of the Curiosity HPC development board, QT7 Xplained Pro extension board from Microchip, and the IR Click Board™ from MikroElektronika. The [IR Receiver and DC Motor Control](#) section is comprised of the Curiosity HPC development board from Microchip, the DC Motor 8 and IR click boards from MikroElektronika. A small DC motor is controlled by the PWM module of the microcontroller.

5.1 Curiosity HPC Development Board

The Curiosity HPC development board is a fully integrated 8-bit development platform. The Curiosity board includes an integrated programmer/debugger and requires no additional hardware to get started. The development board offers several options for user interface, including physical switches, on-board potentiometer and user indication LEDs. A full complement of accessory boards are available via the MikroElektronika mikroBUS™ interface sockets.

Figure 5-1. Curiosity HPC Development Board



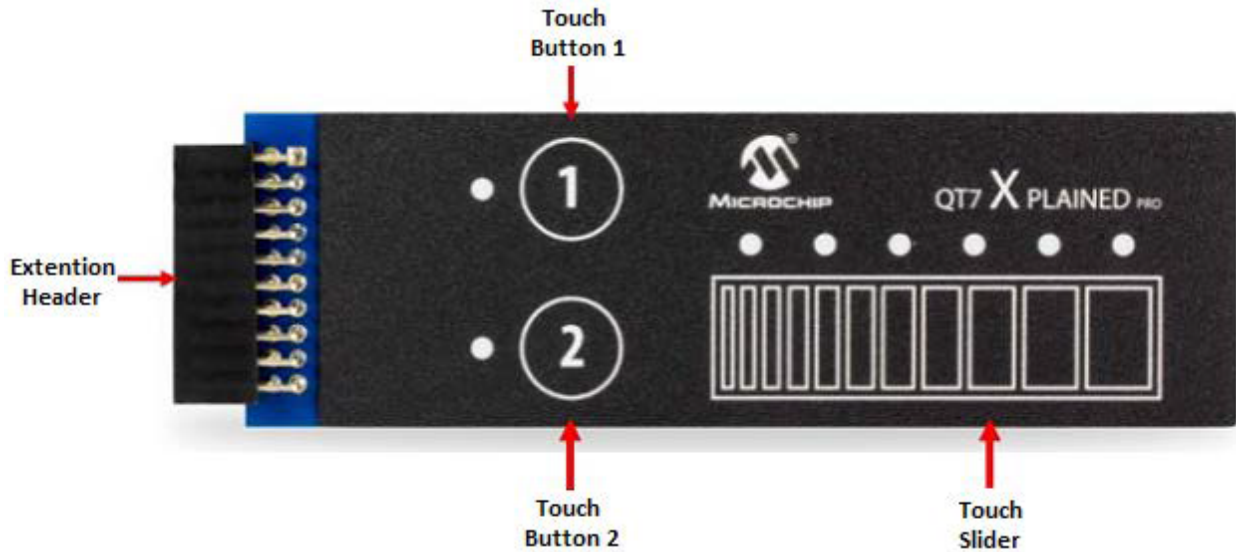
5.2 QT7 Xplained Pro

The QT7 Xplained Pro extension board is designed to explore the touch functions.

Features:

- Two Self-Capacitance Touch Keys/Buttons
- One Self-Capacitance Touch Slider
- Eight LEDs:
 - One LED for each key
 - Six LEDs for the sliders

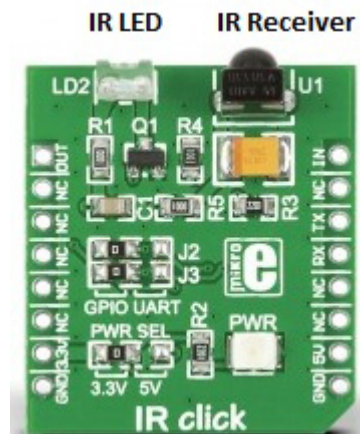
Figure 5-2. QT7 Xplained Pro Extension Board



5.3 IR Click

The IR click is a compact and easy solution for adding IR remote control to the design. It features a TSOP38338 IR receiver module and a QEE113 IR emitting diode. The 38 kHz receiver carrier frequency is recommended for RCM, NEC, RC5, RC6, r-step and XMP codes. The IR click communicates with the target microcontroller via the mikroBUS UART (TX and RX) or AN and PWM lines. Jumpers J2 and J3 are used to choose between these two ways. In this application, AN and PWM signals are used. J1 zero-ohm SMD jumper is used to select between the 3.3V or 5V power supply. The 3.3V option is selected by default.

Figure 5-3. IR Click

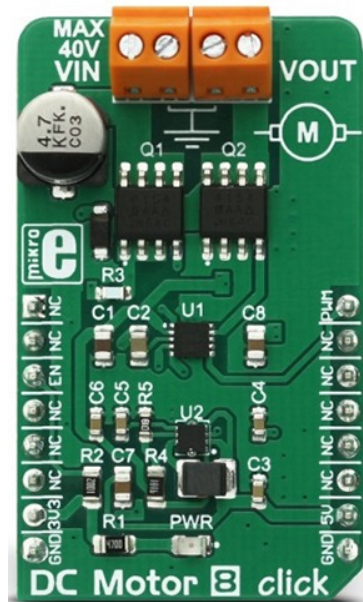


5.4 DC Motor 8 Click

The DC Motor 8 click is a DC motor driver. It can drive simple DC motors with brushes, providing them with a significant amount of current and voltage up to 40V. The DC Motor 8 click relies on the MIC4605, 85V half-bridge MOSFET driver with adaptive dead-time and shoot-through protection. This IC uses the

input on its PWM pin to regulate the switching state of the output MOSFETs. The VIN power terminal used to provide up to 40V of power supply for the DC motor is completely isolated from the driver circuitry. However, to operate correctly, the driver has to provide enough voltage to activate the MOSFETs. For this purpose, the DC Motor 8 click employs a boost converter made of MIC2606 - a 2 MHz boost regulator from Microchip. The boost regulator circuitry provides 12V out of 5V from the mikroBUS, which allows for ideal MOSFET switching conditions, keeping the resistance through the MOSFET ($R_{DS(on)}$) at optimal levels. This DC motor driver can be used when a powerful and reliable DC motor driver is required such as various battery operated hand-held tools, fans, etc.

Figure 5-4. DC Motor 8 Click



5.5 DC Motor

A small DC gear motor with a toy car robot plastic tire wheel is used in the demo. The DC motor specifications are:

- Voltage: DC 3V-6V
- Current: 100 mA-120 mA
- RPM (with tire): 100-240
- Tire Diameter: 65 mm
- Car Speed (M/minute): 20-48
- Motor Size: 70 mm X 22 mm X 18 mm

Figure 5-5. DC Motor with a Toy Car Tire



6. Firmware Overview

6.1 Software Tools

The MPLAB Integrated Development Environment (IDE), compiler and graphical code generator tools are used throughout the application firmware development to provide an easy and hassle-free user experience. The tools used for this demo application are as follows:

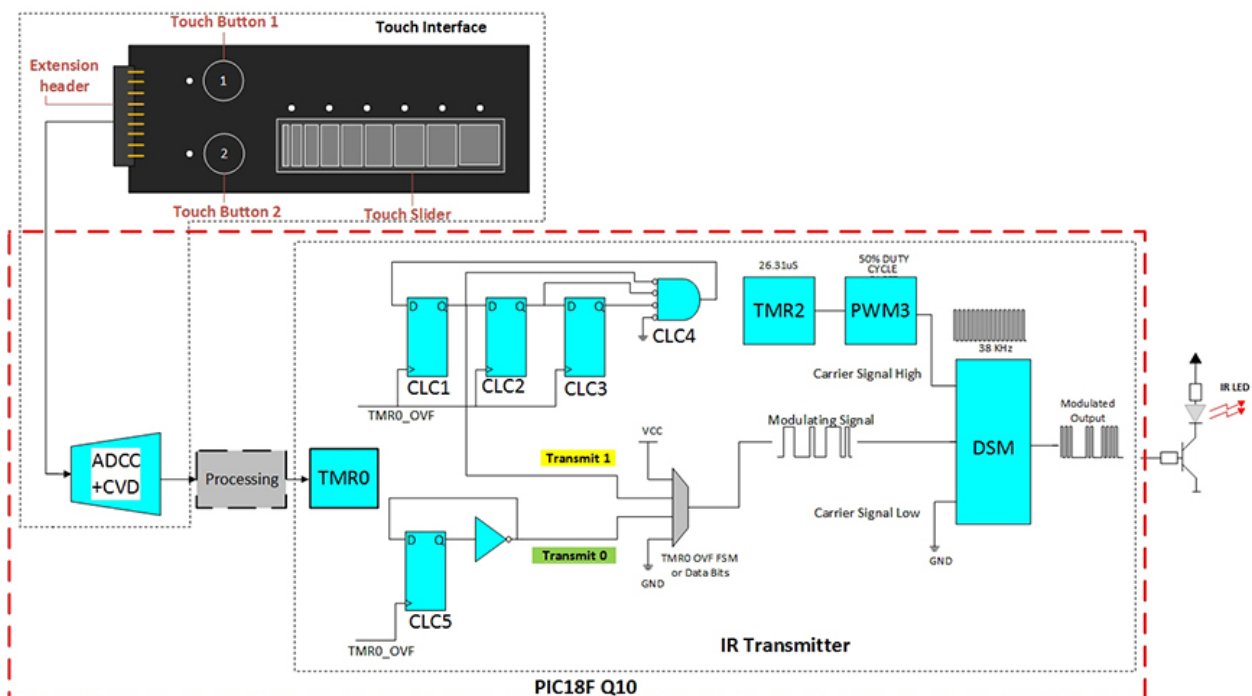
- MPLAB X IDE V5.20
- XC8 Compiler V2.05
- MPLAB Code Configurator (MCC) V3.75
- mTouch Capacitive Sensing Library Module for MCC

Note: For running the demo, the installed tool versions must be the same or later. This example is not tested with the previous versions.

6.2 Touch Interface and IR Transmitter

This application uses a combination of intelligent analog and Core Independent Peripherals (CIPs) to build a touch interface and IR transmitter. For the touch interface, the ADC² along with hardware CVD is used. The IR transmitter uses the NEC IR transmission protocol and is implemented using CIPs such as CLCs, PWM and DSM along with Timer0, as shown in the figure below. The application uses MCC to configure different CIPs and interconnect them in order to generate a modulated signal. The modulated signal is fed to the IR click's IR LED for transmitting a command.

Figure 6-1. Touch Interface and IR Transmitter Block Diagram



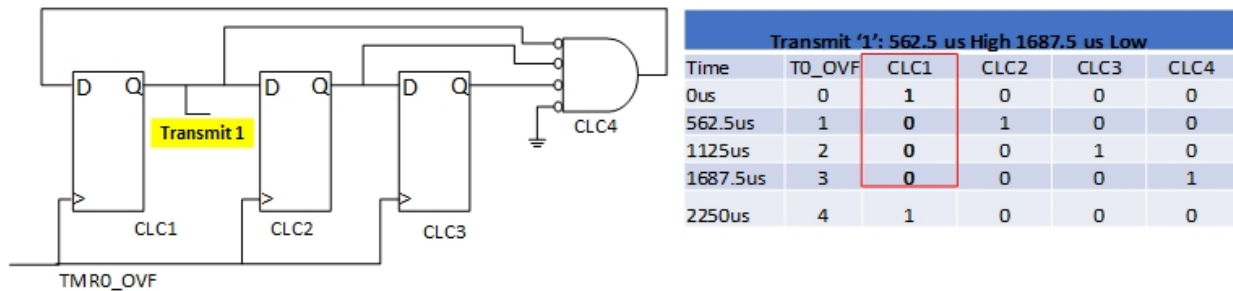
6.2.1 Touch Interface

In the application firmware the touch interface is enabled using the mTouch capacitive sensing library module for MCC, which allows for quick and easy C code generation of capacitive touch button, proximity sensor, and slider/wheel solutions. For each button touch or slider position change event, the corresponding 8-bit command is taken from the look-up table. Then the 32-bit data to be sent over the IR interface, which are comprised of the IR receiver's address, inverse of the address, 8-bit command and inverse of the command, are prepared for that event.

6.2.2 IR Transmitter

The DSM peripheral is used for generating the ON-OFF Keying (OOK) modulated signal transmitted by the IR transmitter. The application decides which signal to be connected to the modulating signal of a DSM, depending on whether to transmit the start sequence or whether the data bit to be sent is logic '1' or logic '0' or end sequence. For data bit transmission, Timer 0 generates an overflow event every 562.5 μ s. If the data bit to be transmitted is logic '1', then as per NEC protocol 562.5 μ s high pulse followed by 1.6875 ms low pulse is generated using CLC1, CLC2, CLC3 and CLC4 peripherals, as shown in the figure below.

Figure 6-2. CLC Configuration to Transmit '1'



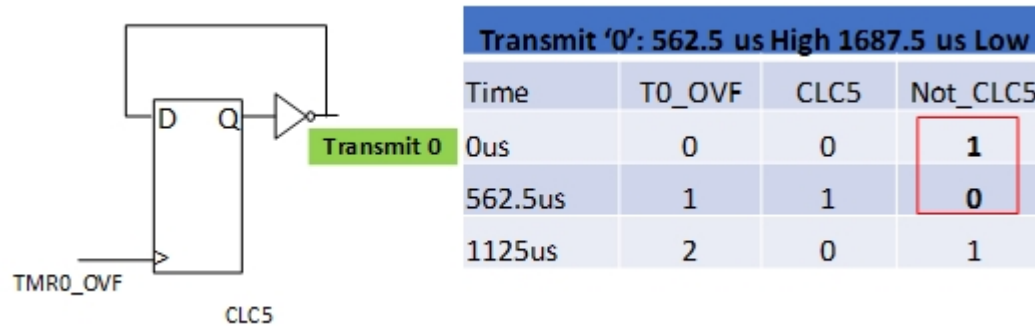
Transmit Logic '1':

- CLC1, CLC2 and CLC3 are configured as D flip-flops.
- CLC4 is configured as an AND gate with four inputs.
- Timer 0 overflow event is configured as a clock for the three D flip-flops.
- Initially, output of CLC1 is set.
- At next Timer 0 overflow, the output of CLC1 D flip-flop becomes low and CLC2 output becomes high.
- At next Timer 0 overflow, the output of CLC1 and CLC2 D flip-flops becomes low and CLC3 output becomes high.
- At next Timer 0 overflow, the output of CLC1, CLC2 and CLC3 D flip-flops becomes low and CLC4 output becomes high. As seen in the table inside the figure above the output of CLC1 is '1,0,0,0' i.e., 562.5 μ s high pulse and 1.6875 ms low pulse.
- Output of CLC1 is connected to DSM modulating signal for transmission of logic '1' data bit.

Transmit Logic '0':

If the data bit to be transmitted is logic '0', then as per NEC protocol 562.5 μ s high pulse followed by 562.5 μ s low pulse is generated using CLC5, as shown in the following figure.

Figure 6-3. CLC Configuration to Transmit '0'



- CLC5 is configured as D flip-flop and the output of CLC5 is negated.
- Timer 0 overflow event is configured as clock for the D flip-flop.
- Initially, output of CLC5 D flip-flop is low and the negated output of CLC5 is high.
- At next Timer 0 overflow, the output of CLC5 D flip-flop becomes high and the negated CLC5 output becomes low.
- As seen in the table inside the figure above the negated output of CLC5 is '1,0' i.e. 562.5 μ s high pulse and 562.5 μ s low pulse.
- Negated output of CLC5 is connected to DSM modulating signal for transmission of logic '0' data bit.

At the end of each data bit transmission, the next data bit to be transmitted is checked. Either combination of CLC1-4 or CLC5 is enabled and connected to a DSM modulating signal depending on whether the next bit to be transmitted is logic '1' or logic '0'.

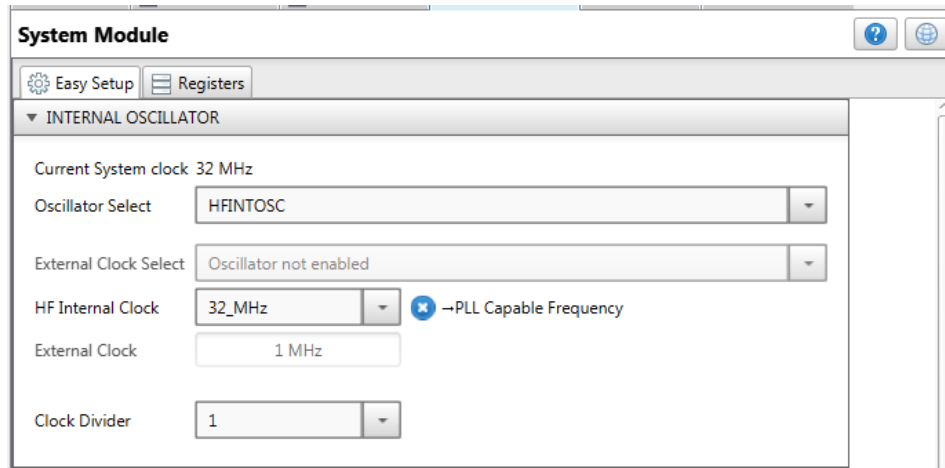
Modulation:

The PWM3 peripheral along with Timer 2 is used for generating the carrier signal of 38 kHz. Whenever the modulating signal is high, the 38 kHz output of the PWM3 module connected to the carrier high signal of a DSM is routed to the DSM output. Whenever the modulating signal is low, the output of a DSM is equal to the carrier low signal that is PWM4 output which is configured as 0% duty to give equivalent signal as ground. The modulated signal that is the output of the DSM is fed to the IR click's IR LED for transmitting a command.

6.2.3 MCU Clock and Peripheral Configurations Using MCC

In the demo firmware for the touch interface and IR transmitter, the High-Frequency Internal Oscillator (HFINTOSC) is used to generate a 32 MHz system clock. The system clock is configured as shown in the figure below, using a system module.

Figure 6-4. System Clock Configuration



Using MCC, the touch sensors and touch parameters are configured as shown in [Figure 6-5](#), [Figure 6-6](#) and [Figure 6-7](#).

Figure 6-5. mTouch Hardware Sensors Configuration

mTouch

Easy Setup

Information

Sensor/Button/Proximity/Slider

Advanced Filtering

Debug

▼ Hardware Sensors

Sensor_ANB0 : RB0 / ANB0

Sensor_ANB2 : RB2 / ANB2

Sensor_ANC4 : RC4 / ANC4

Sensor_AND3 : RD3 / AND3

Sensor_AND2 : RD2 / AND2

▼ Buttons

Button1 : Sensor_ANB0

Button2 : Sensor_ANB2

Slider1_Seg0 : Sensor_AND2

Slider1_Seg1 : Sensor_AND3

Slider1_Seg2 : Sensor_ANC4

Proximities

▼ Sliders

Slider1 : Slider1_Seg0

To create hardware sensors, assign CS pins in the Pin Manager Grid View.

Sensor Sampling Configuration

Idle Sensor Pin State

GND

Dedicated Driven Shield

No driven shield pin enabl

Common Oversampling

32

Scan Waveform Configuration

Auto-Calibration

☒ Enable

Common Pre-Charge Time

10

Common Acquisition Time

5

Additional Sample Capacitance

0 pF

Scan Rate Configuration

Scan Rate Control

Free Running Mode

Scan Rate(ms)

20

Scan Order

Sensor_ANB0 : RB0 / ANB0

Sensor_ANB2 : RB2 / ANB2

Sensor_ANC4 : RC4 / ANC4

Sensor_AND3 : RD3 / AND3

Sensor_AND2 : RD2 / AND2

Up

Down

Figure 6-6. mTouch Button Configuration

mTouch

Easy Setup

Information Sensor/Button/Proximity/Slider Advanced Filtering Debug

▼ Hardware Sensors

- Sensor_ANB0 : RB0 / ANB0
- Sensor_ANB2 : RB2 / ANB2
- Sensor_ANC4 : RC4 / ANC4
- Sensor_AND3 : RD3 / AND3
- Sensor_AND2 : RD2 / AND2

▼ Buttons

- Button1 : Sensor_ANB0**
- Button2 : Sensor_ANB2
- Slider1_Seg0 : Sensor_AND2
- Slider1_Seg1 : Sensor_AND3
- Slider1_Seg2 : Sensor_ANC4

Proximities

▼ Sliders

- Slider1 : Slider1_Seg0

Button Settings Delete Item

Name

Sensor

Threshold

Deviation Scaling

AKS Group

Individual Hysteresis ☐ Enable

Hysteresis Level

Figure 6-7. mTouch Slider Configuration

mTouch

Easy Setup

Information Sensor/Button/Proximity/Slider Advanced Filtering Debug

▼ Hardware Sensors

- Sensor_ANB0 : RB0 / ANB0
- Sensor_ANB2 : RB2 / ANB2
- Sensor_ANC4 : RC4 / ANC4
- Sensor_AND3 : RD3 / AND3
- Sensor_AND2 : RD2 / AND2

▼ Buttons

- Button1 : Sensor_ANB0
- Button2 : Sensor_ANB2
- Slider1_Seg0 : Sensor_AND2
- Slider1_Seg1 : Sensor_AND3
- Slider1_Seg2 : Sensor_ANC4

Proximities

▼ Sliders

- Slider1 : Slider1_Seg0**

Slider Settings Delete Item

Name

Slider Type

Start Segment

Contact Size Threshold

Position Hysteresis

Resolution(bit)

Deadband(%)

AKS Group

The following peripherals of the PIC18 Q10 MCU are configured using MCC. The PWM3 peripheral along with Timer 2 is used for generating the carrier signal of 38 kHz. The Timer 2 configuration is shown in the following figure.

Figure 6-8. Timer 2 Configuration

TMR2
?

Easy Setup
Registers

☒ Enable Timer

Control Mode

Roll over pulse

Ext Reset Source

T2CKIPPS pin

Start/Reset Option

Software control

Timer Clock

Clock Source

FOSC/4

☐ Enable Clock Sync

Clock Frequency

32.768 kHz

Polarity

Rising Edge

Prescaler

1:1

☐ Enable Prescaler O/P Sync

Postscaler

1:1

Timer Period

Timer Period
 125 ns ≤

26.16 us

 ≤ 32 us

Actual Period 26.125 us (Period calculated via PR Register value)

Software Settings

☐ Enable Timer Interrupt

The PWM3 and PWM4 configuration is shown in the figure below. PWM4 is connected to carrier low signal of the DSM to generate low output. PWM4 duty cycle is set as 0%.

Figure 6-9. PWM3 and PWM4 Configuration

PWM3

Easy Setup

Registers

Hardware Settings

☒ Enable PWM

Select a Timer :

Timer2

Duty Cycle

Duty Cycle

50

 %

PWMDC Value 417

PWM Parameters

PWM Polarity

active_hi

PWM Period 26.125 us

PWM Frequency 38.27751 kHz

PWM Resolution 9 Bits

PWM4

Easy Setup

Registers

Hardware Settings

☒ Enable PWM

Select a Timer :

Timer2

Duty Cycle

Duty Cycle

0.0

 %

PWMDC Value 0

PWM Parameters

PWM Polarity

active_hi

PWM Period 26.125 us

PWM Frequency 38.27751 kHz

PWM Resolution 9 Bits

CLC1, CLC2, CLC3 and CLC4 are configured together for transmitting bit as logic '1'. CLC5 is configured for transmitting bit as logic '0'. The CLCs are configured as shown in [Figure 6-10](#), [Figure 6-11](#), [Figure 6-12](#), [Figure 6-13](#) and [Figure 6-14](#).

Figure 6-10. CLC1 Configuration

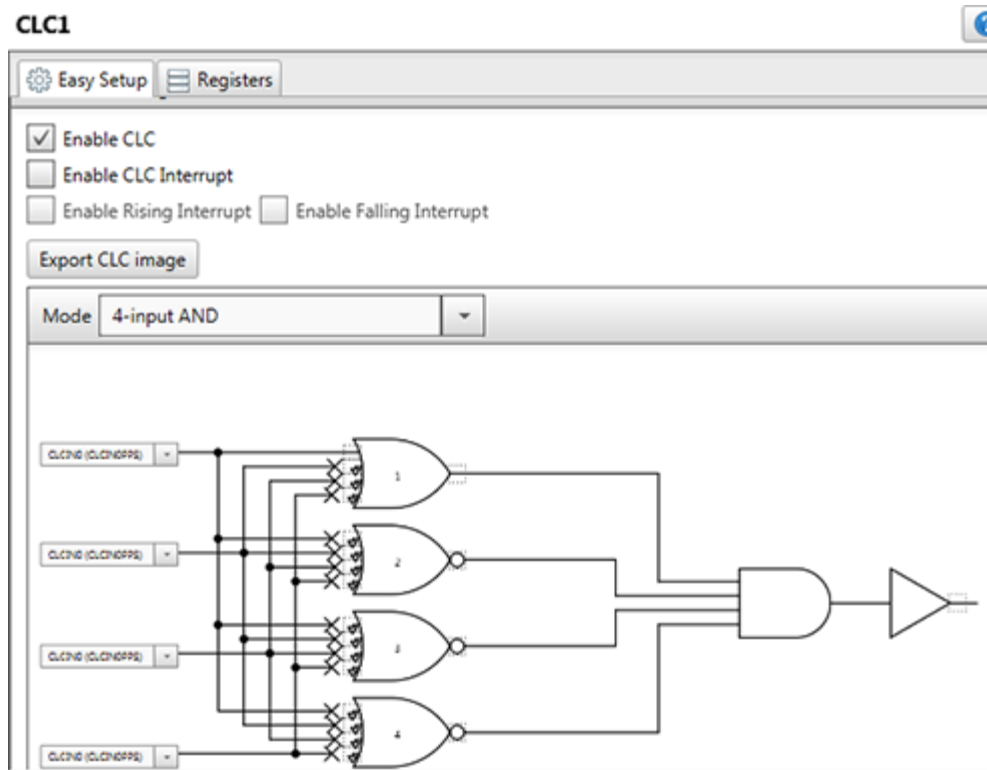


Figure 6-11. CLC2 Configuration

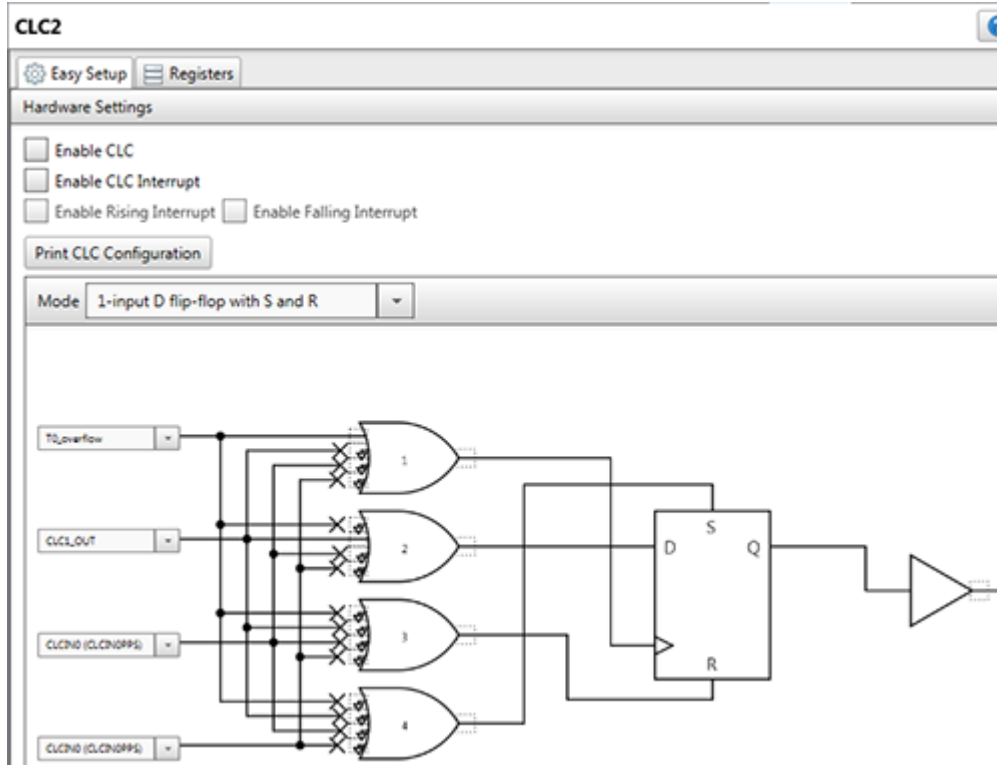


Figure 6-12. CLC3 Configuration

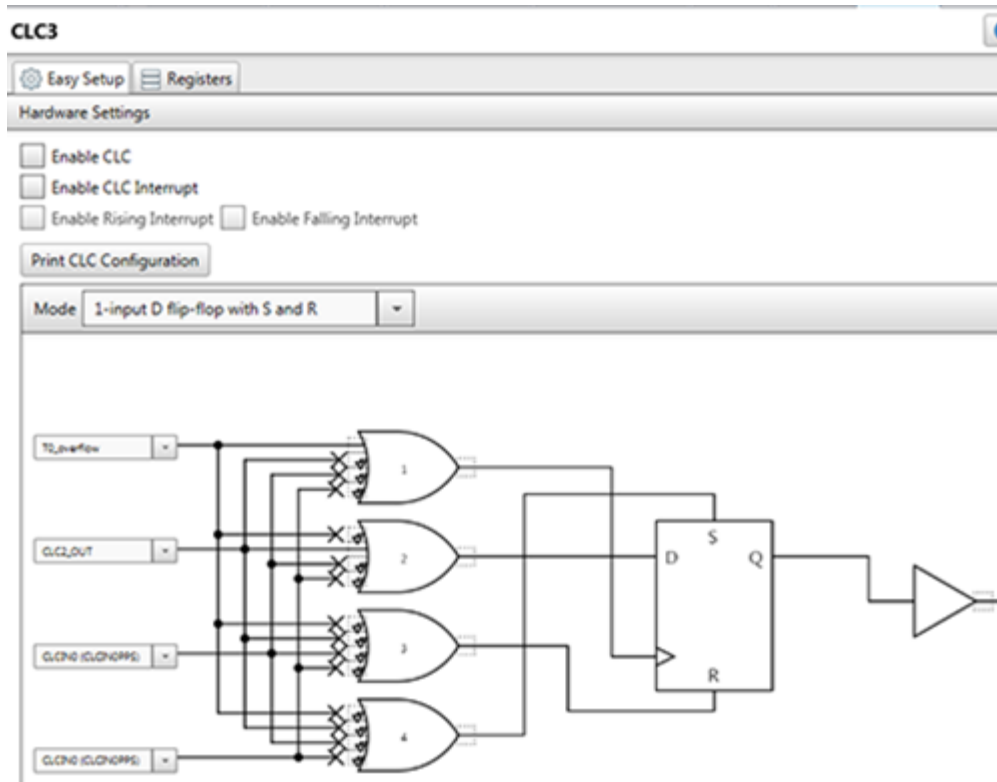


Figure 6-13. CLC4 Configuration

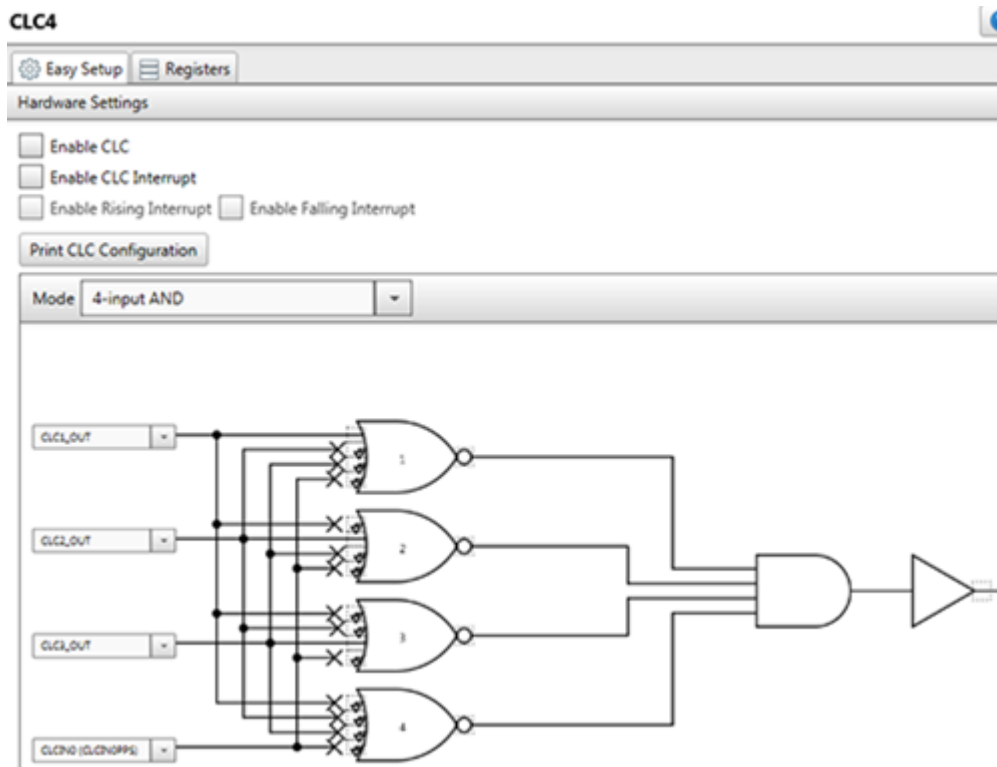
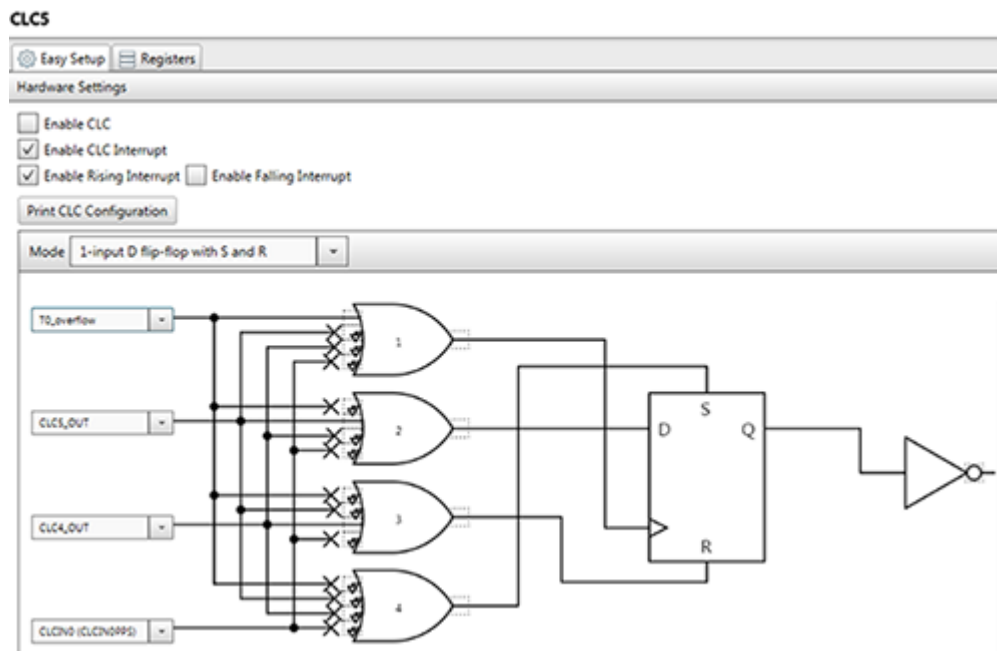


Figure 6-14. CLC5 Configuration



The PWM3 module generates a 38 kHz carrier signal. The DSM module takes the 38 kHz carrier signal from the PWM3 module as one of the inputs and generates a modulated waveform. The modulated waveform needs to be fed to the IR LED for transmission. The DSM configuration is shown in the following figure.

Figure 6-15. DSM Configuration

The screenshot shows the 'DSM' configuration window. It has two tabs: 'Easy Setup' and 'Registers'. Under 'Hardware Settings', the following options are checked: 'Enable Modulator' and 'Invert Output Polarity'. The 'Modulation Source' dropdown is set to 'MDBIT'. Below this, there are two sections: 'High Carrier' and 'Low Carrier'. In the 'High Carrier' section, the 'Carrier' dropdown is set to 'PWM3_out', and the 'Invert Polarity' and 'Synchronize Carrier' options are unchecked. In the 'Low Carrier' section, the 'Carrier' dropdown is set to 'PWM4_out', and the 'Invert Polarity' and 'Synchronize Carrier' options are unchecked.

6.2.4 Firmware Operation

The following steps are needed for transmitting a command.

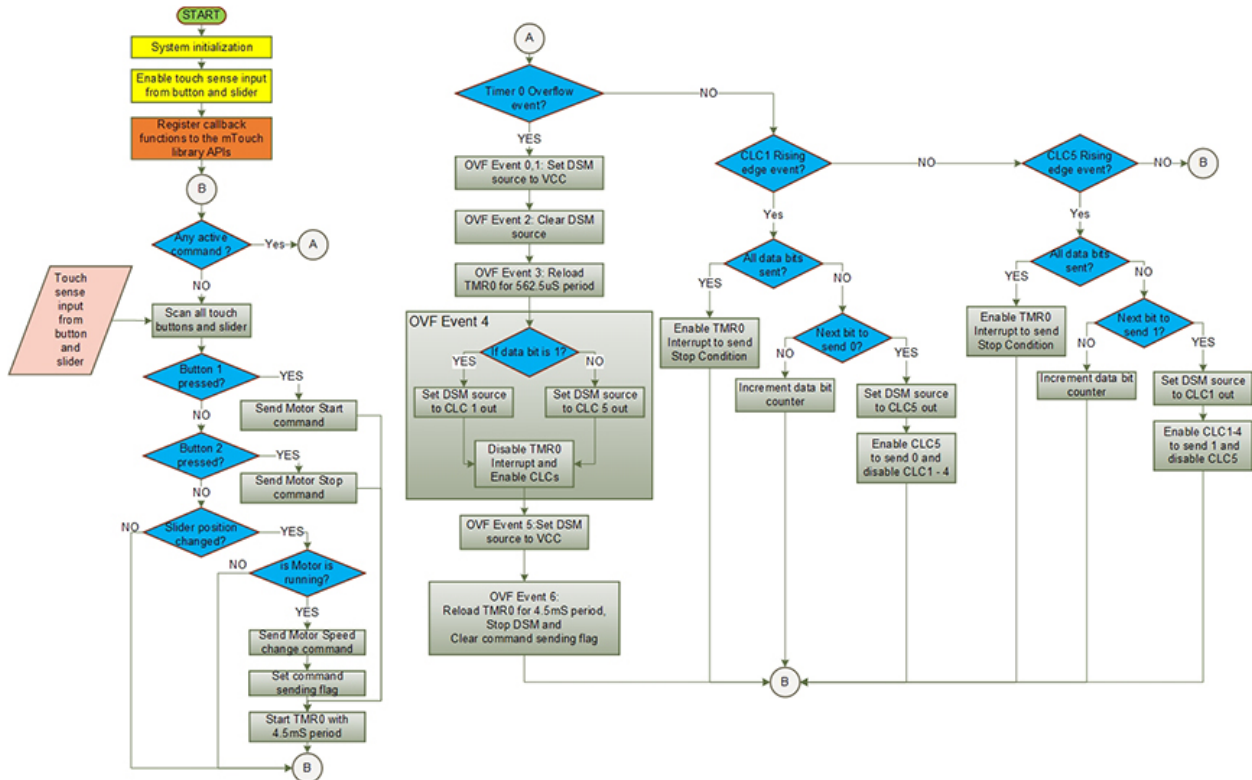
1. Command generation from a button touch event or slider position change event:

- After the button touch or slider position change event on the QT7 Xtension Pro board, generate one byte of command to be transmitted.
 - Reverse the bits in the command byte so that LSb will be transmitted first as per NEC protocol.
 - Take the address of the receiver and reverse the bits for transmitting LSb first.
 - Generate a 32-bit data value (address, inverse of address, command, inverse of command) for transmission.
2. Transmit command:
- Start Timer 0, to overflow every 4.5 ms.
3. Transmit start:
- For the first overflow interrupt, set the DSM modulating signal so that a pulse burst of 38 kHz from PWM3 peripheral is connected to output on TX pin for the start sequence of 9 ms. Timer 0 generates overflow interrupt for two consecutive times, so that a 38 kHz pulse burst is transmitted for 9 ms.
 - Make the DSM modulating signal low and wait for one more timer 0 overflow interrupt so that the TX pin is connected to ground for 4.5 ms, which completes the start pulse.
4. Transmit 32-bit data (IR receiver address, inverse of address, command and inverse of command):
- After start pulse, change the timer period value to transmit 32-bit data at 562.5 μ S period. Change timer 0 pre-scaler to 1:32 to have Timer 0 overflow pulse of 4 μ S. This makes sure that Timer 0 overflow signal connected as clock to CLCs is in low state when the CLCs are initialized and ensures proper operation of the CLCs.
 - Disable TMR0 interrupt. Further interrupts are handled in CLC ISR.
5. Transmit bit logic '1' or logic '0' using CLCs:
- CLC1, CLC2, CLC3 and CLC4 are configured to transmit logic '1'.
 - CLC5 is configured to transmit logic '0'.
 - Extract the first bit from 32-bit data value and transmit logic '1' or logic '0' accordingly.
 - Once the logic 1/0 is transmitted, CLC interrupt is generated (CLC1 interrupt for logic '1' and CLC5 interrupt for logic '0').
 - Check if all the 32 bits are transmitted.
 - If all bits are not transmitted, check if the next bit to be transmitted is logic 1/0.
 - Case 1: If logic '0' is transmitted before and the next bit is '0', no action is needed; let the state machine transmit one more bit of logic '0'.
 - Case 2: If logic '0' is transmitted before and the next bit is '1', disable CLCs for transmitting logic '0' and enable CLCs to transmit logic '1'.
 - Case 3: If logic '1' is transmitted before and the next bit is '1', no action is needed; let the state machine transmit one more bit of logic '1'.
 - Case 4: If logic '1' is transmitted before and the next bit is '0', disable CLCs for transmitting logic '1' and enable CLCs to transmit logic '0'.
6. Transmit end:
- If all the 32 bits are transmitted, enable the Timer 0 interrupt, make the DSM modulating signal high for transmitting the Stop bit.
7. Reinitialize for next command transmission:

- Reinitialize the timer to generate overflow interrupt for every 4.5 ms and wait for the next command to be transmitted.

6.2.5 Application Flowchart for Touch Interface and IR Transmitter

Figure 6-16. Application Flowchart for Touch Interface and IR Transmitter



6.2.6 Program and Data Memory Requirements

The application firmware's program and data memory requirements for various compiler optimization levels are given below.

Table 6-1. Touch Interface and IR Transmitter Application Memory Requirements

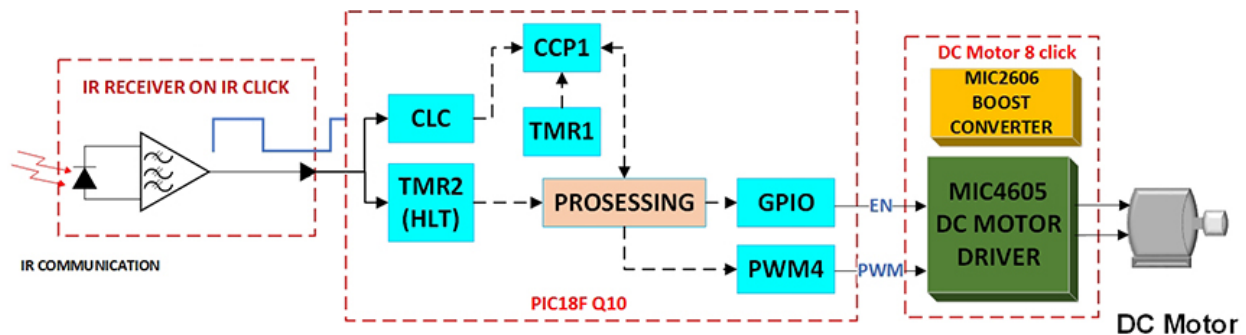
Sr. No.	Optimization Level	Program Memory (in bytes)	Data Memory (in bytes)
1	O0	11927	353
2	O1	11927	353
3	O2	11495	349
4	O3	10358	337
5	S	9206	337

6.3 IR Receiver and DC Motor Control

This application uses a combination of CIPs to build an IR receiver with the NEC IR protocol and control a small DC motor. The IR receiver is implemented using a Capture Compare Peripheral (CCP) along with Timer1, CLC and Timer 2 (HLT) as shown in the following figure. The signals to control the DC motor are provided to the DC motor driver on the DC Motor 8 click board. This application uses the MCC to

configure different CIPs and interconnect them to control the DC motor according to commands received over the IR interface.

Figure 6-17. IR Receiver and DC Motor Control Block Diagram



6.3.1 IR Receiver

The IR photo diode present on the IR click board demodulates the received IR data and removes the 38kHz carrier signal. The demodulated data is connected to the microcontroller port pin. Whenever the first falling edge is detected on the port pin, an HLT (Timer 2) in Monostable mode will start automatically. Upon timer overflow interrupt after few ms it is confirmed whether the data line is still low, and the detected falling edge is not due to electrical noise. CCP is used to capture timing of incoming data frame. CCP can be used with either port B or port C of the microcontroller and the demodulated data from IR click is connected to port pin RA1. So CLC is used as buffer or an interconnecting element to connect port pin RA1 internally to port pin RC0. RC0 is used as input to CCP. CCP peripheral along with Timer 1 is used for start/end sequence detection and detection of logic '1' and logic '0' data bits. The length of all the 32 data bits is captured using CCP and is stored in a data buffer. The command is decoded from the received 32 bits of data.

6.3.2 DC Motor Control

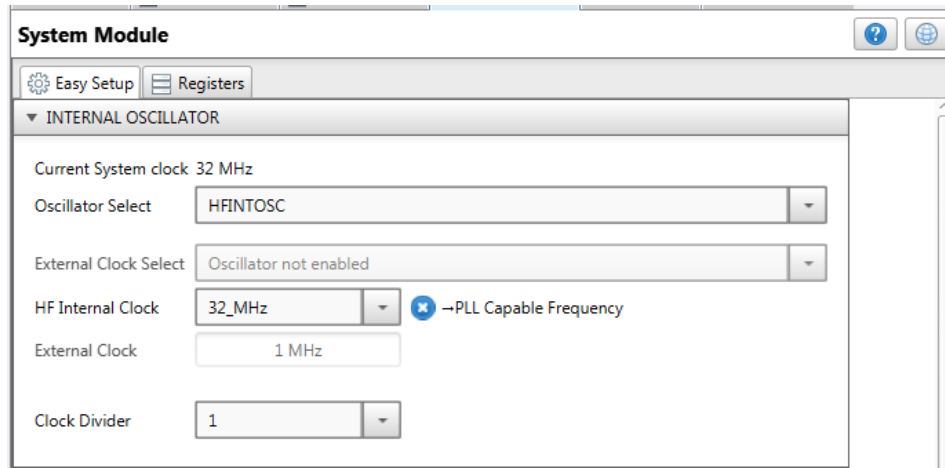
Corresponding control action is taken after decoding the command received by the IR receiver. Start and Stop states of the DC motor are controlled using the EN signal connected to the DC Motor 8 click board. The speed of the DC motor is controlled by varying the duty cycle of the PWM peripheral. PWM output is also connected to the DC Motor 8 click board. DC Motor 8 click board includes the MIC4605 half-bridge DC motor driver. Various commands and corresponding control actions are described in the section [Firmware Operation](#).

This application uses the MCC to configure different CIPs and interconnect them to control the DC motor according to commands received over the IR interface.

6.3.3 MCU Clock and Peripheral Configurations Using MCC

In the demo firmware for IR Receiver and DC motor control, the High-Frequency Internal Oscillator (HFINTOSC) is used to generate a 32 MHz clock.

Figure 6-18. System Clock Configuration



The following peripherals of the PIC18 Q10 MCU are configured using MCC. Timer 2 is used in monostable configuration as shown in below figure for detecting valid falling edge on port pin RA1.

Figure 6-19. Timer 2 Configuration

TMR2

Easy Setup

Registers

☒ Enable Timer

Control Mode

Roll over pulse

Ext Reset Source

T2CKIPPS pin

Start/Reset Option

Software control

Timer Clock

Clock Source

FOSC/4

☐ Enable Clock Sync

Clock Frequency

32.768 kHz

Polarity

Rising Edge

Prescaler

1:1

☐ Enable Prescaler O/P Sync

Postscaler

1:1

Timer Period

Timer Period

125 ns ≤ 26.16 us ≤ 32 us

Actual Period 26.125 us (Period calculated via PR Register value)

Software Settings

☐ Enable Timer Interrupt

CCP1 along with Timer 1 is used for capturing the commands received by the IR receiver.

Figure 6-20. Timer 1 Configuration

TMR1
?
globe

Easy Setup
Registers

Hardware Settings

☐ Enable Timer

Timer Clock

Clock Source FOSC/4

External Frequency 32.768 kHz

Prescaler 1:2

☒ Enable Synchronization

Timer Period

Timer Period 250 ns ≤ 2 ms ≤ 16.384 ms

Period count 0x0 ≤ 0xE0C0 ≤ 0xFFFF

Calculated Period 2 ms

☐ Enable 16-bit read

☐ Enable Gate

☐ Enable Gate Toggle Gate Signal Source T1G_pin

☐ Enable Gate Single-Pulse mode Gate Polarity low

☐ Enable Timer Interrupt

☐ Enable Timer Gate Interrupt

Software Settings

Callback Function Rate 0 x Time Period = 0.0 ns

Figure 6-21. CCP1 Configuration

CCP1
?
globe

Easy Setup
Registers

Hardware Settings

☐ Enable CCP

CCP Mode Capture

Select Timer Timer 1

Input Signal CCP1 pin

Capture Mode Falling edge

Please refer to the assigned timer for captured values.

☒ Enable CCP Interrupt

The speed of the DC motor is controlled by varying duty cycle of the PWM generated using the Timer 4 and PWM3 peripherals.

Figure 6-22. Timer 4 Configuration

TMR4
?

Easy Setup
Registers

☒ Enable Timer

Control Mode

Roll over pulse

Ext Reset Source

T4CKIPPS pin

Start/Reset Option

Software control

Timer Clock

Clock Source

FOSC/4

☐ Enable Clock Sync

Clock Frequency

32.768 kHz

Polarity

Rising Edge

Prescaler

1:1

☐ Enable Prescaler O/P Sync

Postscaler

1:1

Timer Period

Timer Period
 125 ns ≤

25 us

 ≤ 32 us

Actual Period
 25 us (Period calculated via PR Register value)

Software Settings

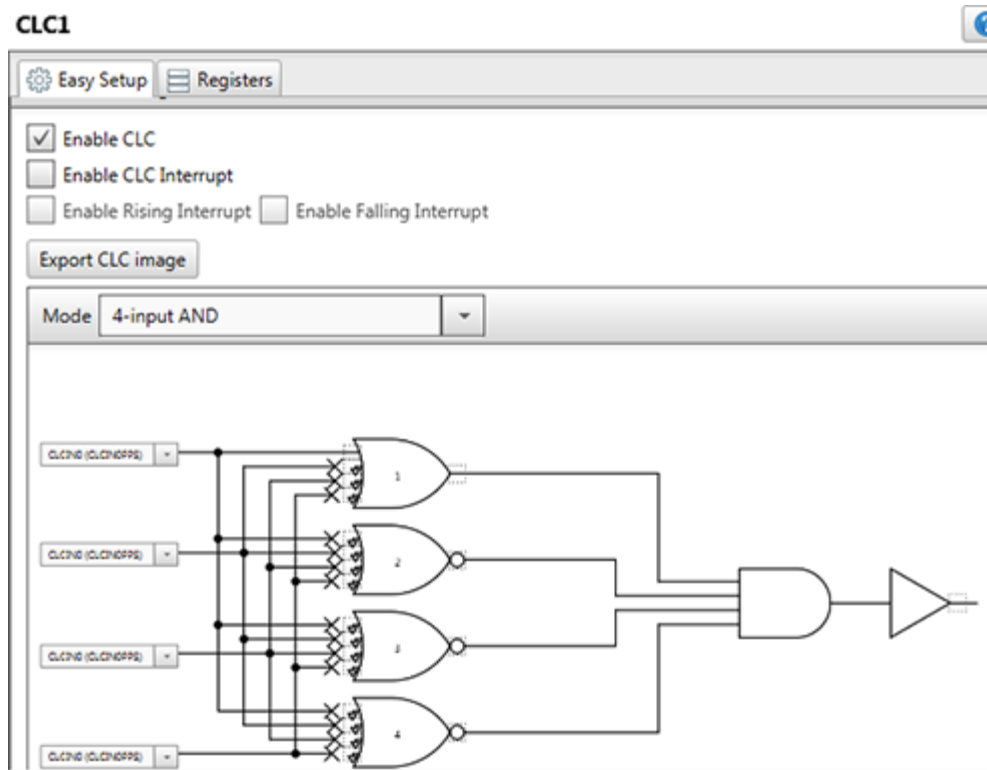
☐ Enable Timer Interrupt

Figure 6-23. PWM3 Configuration

The screenshot shows a software configuration window titled "PWM3". At the top right are two icons: a question mark and a globe. Below the title bar are two tabs: "Easy Setup" (selected) and "Registers". The "Easy Setup" tab contains a "Hardware Settings" section with a checked "Enable PWM" checkbox and a "Select a Timer" dropdown menu set to "Timer4". Below this is a "Duty Cycle" section with a "Duty Cycle" input field set to "0.0" and a percentage sign, and a "PWMDC Value" set to "0". At the bottom is a "PWM Parameters" section with a "PWM Polarity" dropdown set to "active_hi", and three read-only fields: "PWM Period" (25 us), "PWM Frequency" (40 kHz), and "PWM Resolution" (9 Bits).

The IR received data coming from the IR click board is connected to pin RA1 of the microcontroller. CCP is used to capture received data. Only port B or C can be used as input to the CCP. So, RC0 is configured as the CCP capture input pin. CLC can be used as buffer to connect port pin RA1 internally to port pin RC0. CLC1 configuration as interconnecting element is shown in below figure.

Figure 6-24. CLC1 Configuration



6.3.4 Firmware Operation

The data received over IR is made available on pin RA1 of the MCU, which is connected to the IR click's AN pin. By default, the status of the pin is logic high. Whenever data is received, the pin status becomes logic low.

IR frame detection:

- Timer 2 (HLT) can be started on an external event when used in Monostable mode. The Timer is configured to start automatically when a falling edge is present on RA1.
- The Timer period is set as 2ms. Using Timer 2, check whether the reported falling edge is proper and because of start of the IR frame or the change is momentary and is due to the electrical noise.
- After 2 ms, upon Timer 2 overflow event, check if RA1 status is still logic low. If RA1 is logic low, then only set the falling edge detected flag and reset Timer's ON bit.
- If the falling edge detected flag is set then start Timer 1, enable CCP1 module, and capture interrupt for Start condition detection and for receiving the IR command. Disable Timer overflow interrupt now, as all the interrupts related to falling edge in the rest of the IR frame are handled by the CCP1.

Start sequence detection in IR frame:

- Upon CCP1 capture event, check if start sequence is received with proper length (13.5 ms +/- bit error). If the capture length is proper, then capture 32 bit data from the IR frame.

32-bit data reception using CCP:

- Upon each CCP1 capture event check whether the length of the received bits (i.e., logic '1' or logic '0') is appropriate. Store the captured value corresponding to each few the 32 data bits in the buffer

variable. The timer count or capture count corresponding to the Start condition, logic '1' and logic '0' data bit, is defined in the header file `application.h`.

- While receiving data, anytime the captured value goes out of range, the IR transaction is aborted and the registers and variables are reinitialized for receiving the next command.
- If all the bits/edges are received properly, Timer 1 is stopped and disabled, CCP module is disabled, and Timer 2 (HLT) in Monostable mode is enabled for the next IR frame detection.

Decoding command:

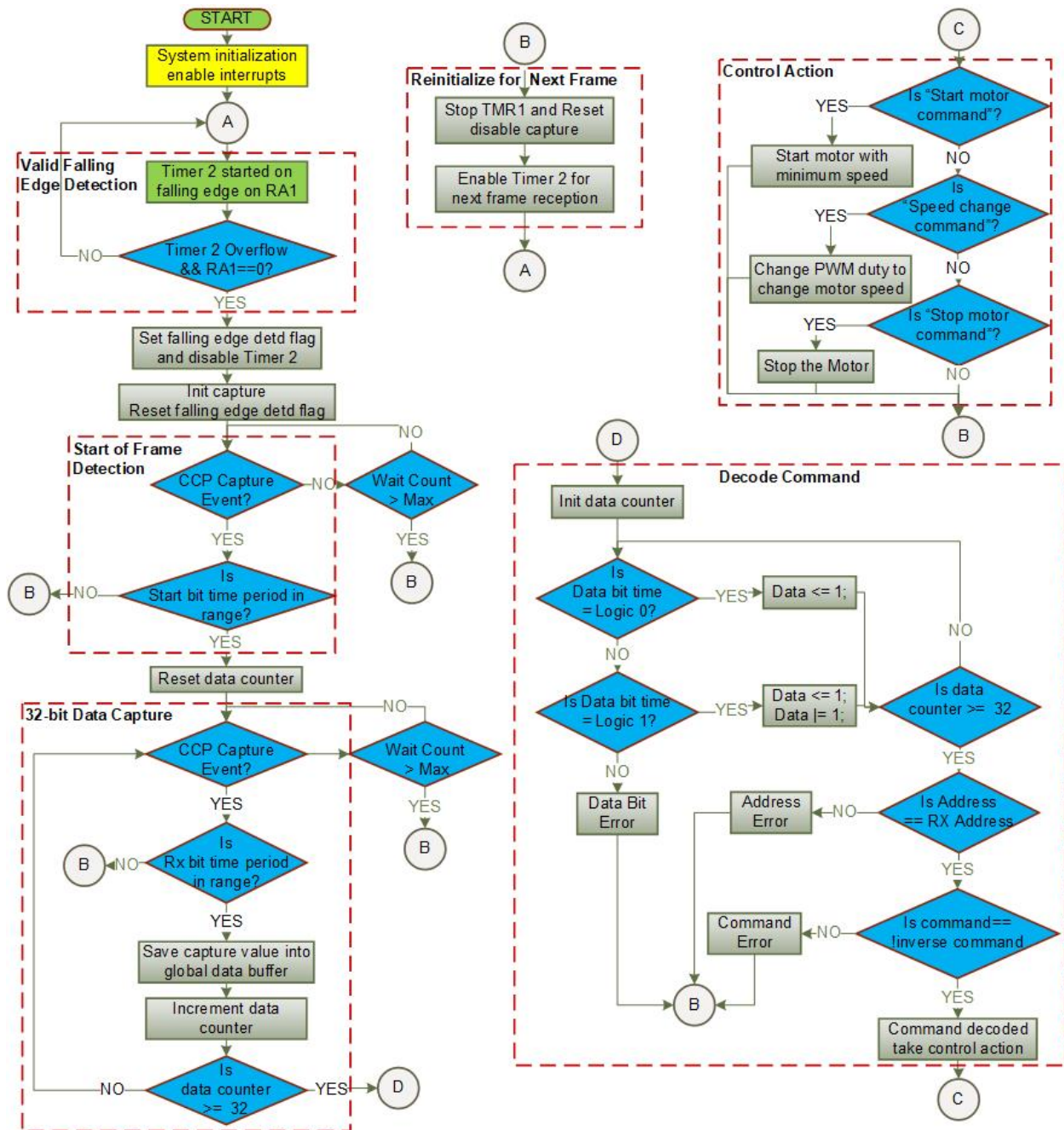
- After capturing the 32 bit data bits from the IR frame, the command is decoded/extracted from the received data.
- Contents of the capture buffer are checked for the correct IR receiver address, inverse of address, command byte and its inverse.
- If the received IR receiver address and address inverse, command and command inverse are matching, then the command is formed by reversing the bits in the received command byte. As in the NEC IR protocol, LSb is the first bit to be transmitted and received.
- Corresponding control action is taken for the command received.

Commands and control action:

- Command #1: Start the motor with minimum speed, having 37.5% PWM duty cycle.
- Command #2: Stop the motor.
- Command #81: Speed change command. Change motor speed to 0% PWM duty cycle.
- Command #82: Speed change command. Change motor speed to 37.5% PWM duty cycle.
- Command #83: Speed change command. Change motor speed to 45% PWM duty cycle.
- Command #84: Speed change command. Change motor speed to 52.5% PWM duty cycle.
- Command #85: Speed change command. Change motor speed to 60% PWM duty cycle.
- Command #86: Speed change command. Change motor speed to 67.5% PWM duty cycle.
- Command #87: Speed change command. Change motor speed to 75% PWM duty cycle.

6.3.5 Application Flow

Figure 6-25. Application Flowchart for IR Reception and DC Motor Control



6.3.6 Program and Data Memory Requirements

The application firmware's program and data memory requirements for various compiler optimization levels are given in the following table.

Table 6-2. IR Receiver and DC Motor Control Application Memory Requirements

Sr. No.	Optimization Level	Program Memory (in bytes)	Data Memory (in bytes)
1	O0	1718	106
2	O1	1718	106
3	O2	1662	104
4	O3	1422	101
5	S	1422	101

7. Demo Operation

7.1 Demo Setup for the Touch Interface and IR Transmitter

- Connect the IR click board in mikroBUS slot 2 of the Curiosity HPC board.
- Connect the QT7 Xplained Pro extension board to jumper J11 on the Curiosity HPC board using a custom wire harness. The required pin connections between the Curiosity HPC board and the QT7 Xplained Pro extension board are as shown in:

Table 7-1. Connections Between QT7 Extension Pro and Curiosity HPC Development Board

Sr. No.	Pin No. of Header 1 - QT7 Extension	Signal Name of Pins on Header 1 - QT7 Extension	Header J11 of Curiosity HPC Board	Signal Description	IN/OUT ⁽¹⁾
1	2	GND	GND	Ground	PWR
2	4	Y-LINE-1	RB0	Y-line 1: Connected to Button 1	IN
3	5	LED0	RC6	Touch Status LED for Slider	OUT
4	6	LED6	RB1	Touch Status LED for Button 1	OUT
5	7	Y-LINE-2	RC4	Y-line 2: Connected to Slider	IN
6	8	Y-LINE-3	RD3	Y-line 3: Connected to Slider	IN
7	9	Y-LINE-4	RD2	Y-line 4: Connected to Slider	IN
8	10	Y-LINE-0	RB2	Y-line 4: Connected to Button 2	IN
9	11	LED7	RB3	Touch Status LED for Button 2	OUT
10	12	LED1	RC7	Touch Status LED for Slider	OUT
11	15	LED2	RD4	Touch Status LED for Slider	OUT
12	16	LED 3	RD5	Touch Status LED for Slider	OUT
13	17	LED 4	RD6	Touch Status LED for Slider	OUT
14	18	LED 5	RD7	Touch Status LED for Slider	OUT
15	19	GND	GND	Ground	PWR
16	20	V _{CC}	TV _{DD}	Target Supply Voltage	PWR

Note: (1) Direction IN/OUT of the signal is with respect to the MCU.

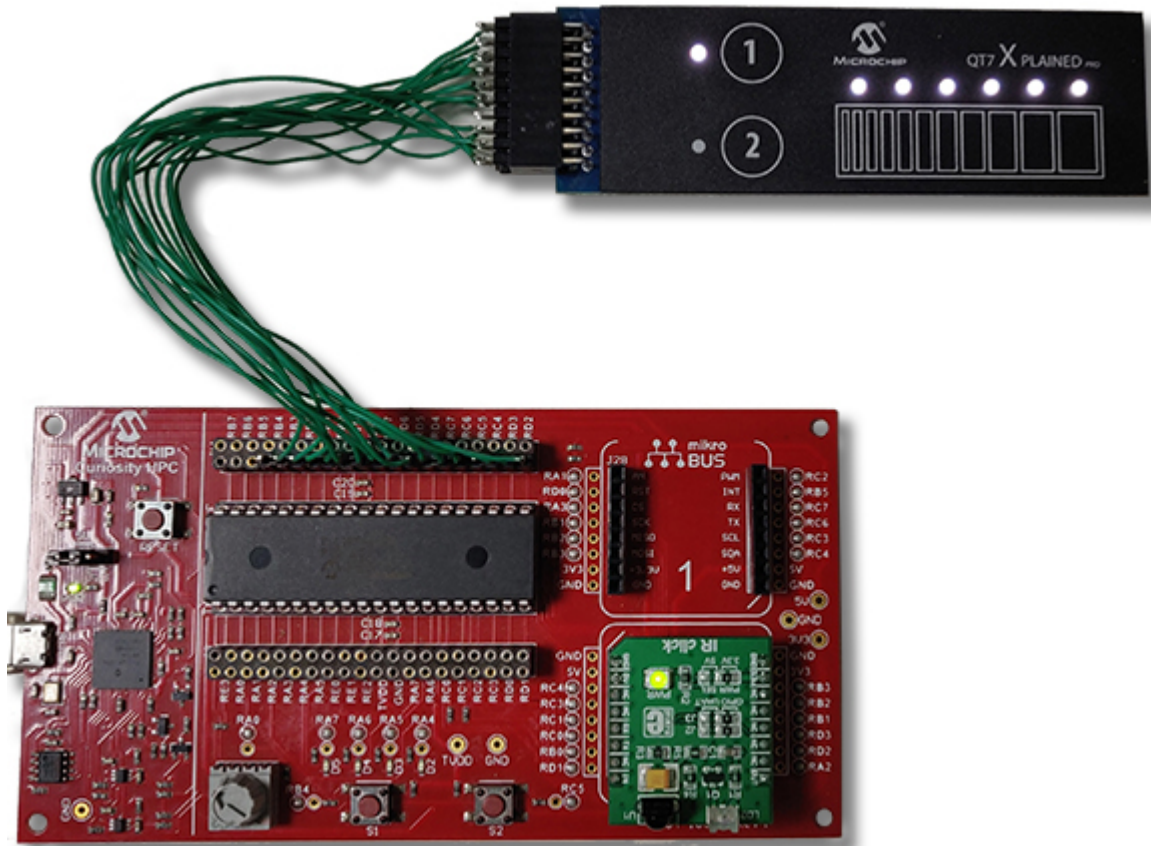
- Other port pins used in the application are described in below table.

Table 7-2. Port Pin Mapping

Sr. No.	MCU Port pin #	Signal Description	IN/OUT
2	RD1	DSM OUT Connected to IR LED on IR_CLICK with PWM pin	OUT
3	RA4	LED_D2 on Curiosity HPC	OUT

.....continued			
Sr. No.	MCU Port pin #	Signal Description	IN/OUT
4	RA5	LED_D3 on Curiosity HPC	OUT
5	RA6	LED_D4 on Curiosity HPC	OUT
6	RA7	LED_D5 on Curiosity HPC	OUT

Figure 7-1. Demo Setup Touch Interface with IR TX



- After making the hardware connections as shown in the figure above, power on the board with a micro-USB cable. Build the demo firmware and load the generated `hex` file to the PIC18 Q10 MCU.
- When the DC motor is stalled, button 1 is used to send the desired IR command to start the DC motor. Indication LED for button 1 will turn ON, and the indication LED for button 2 will turn OFF.
- While the DC motor is running, button 2 is used to send the desired IR command to stop the DC motor. The indication LED for button 2 will turn ON, and the indication LED for button 1 will turn OFF.
- If the motor is in Running state, then the slider is used to send commands to change (increase/decrease) the motor speed.

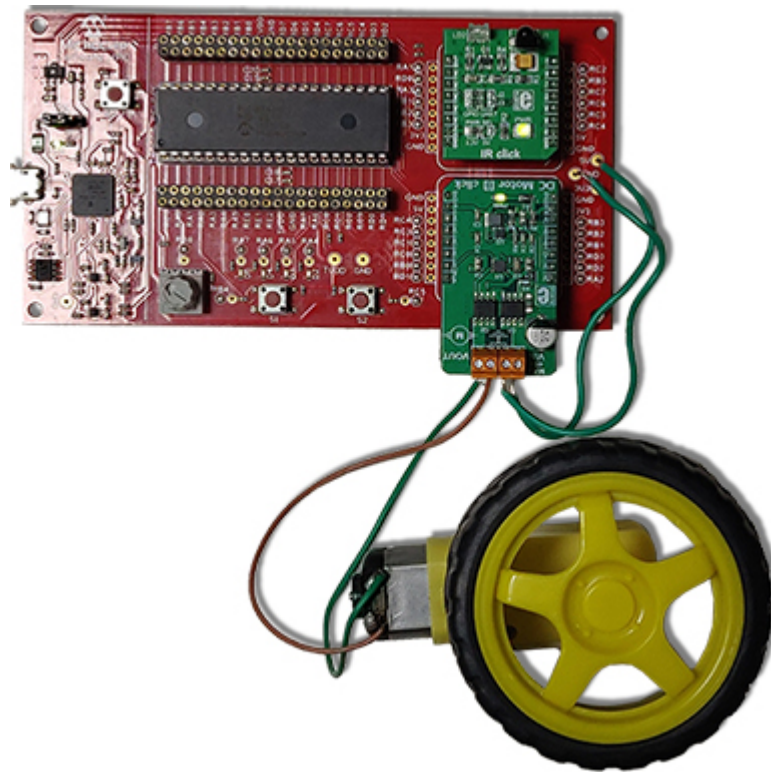
7.2 Demo Setup for the IR Receiver and DC Motor Control

- Connect the IR click board in mikroBUS slot 1 of the Curiosity HPC board.
- Connect the DC Motor 8 click board in mikroBUS slot 2 of the Curiosity HPC board.
- Connect the DC motor to V_{OUT} terminals on DC Motor 8 click board.
- Connect V_{IN} motor power on DC Motor 8 click board to 5V on the Curiosity HPC board.
- Connect PGND on the DC Motor 8 click board to GND on the Curiosity HPC board.
- MCU port pins used in the application are described in the table below.

Table 7-3. MCU Port Pin Mapping

Sr. No.	MCU Port pin #	Signal Name	Signal Description	IN/OUT
1	RA1	IR_RX	IR received signal	IN
2	RC0	CCP1	Capture input	IN
3	RD3	EN	Enable input for MIC4605 driver. Logic high on the enable pin results in normal operation and MIC4605 device enters Shutdown mode with a logic low applied to enable pin.	OUT
4	RD1	PWM3	PWM for motor speed control	OUT
5	RA4	LED_D2	LED_D2 on Curiosity HPC	OUT
6	RA5	LED_D3	LED_D3 on Curiosity HPC	OUT
7	RA6	LED_D4	LED_D4 on Curiosity HPC	OUT
8	RA7	LED_D5	LED_D5 on Curiosity HPC	OUT

Figure 7-2. Demo Setup DC Motor Control with IR RX



- After making the above hardware connections, power up the board with the micro-USB cable. Build the demo firmware and load the generated `hex` file to the PIC18 Q10 MCU.
- Control action is taken based on the commands received over IR communication from the touch interface and IR transmitter board.
- Using button 1 on the IR transmitter board, the motor can be started. Using button 2, it can be stopped, and the speed of the motor can be changed using slider.

8. The Possible Application Use Cases

This application targets simple, real-time control applications. The possible application use cases are:

- Home appliances: Controlling home appliances using an IR remote is basic to home automation. It can be used to control home appliances around the house like an exhaust fan, ceiling fan, table fan, coffee maker, printer, speaker, TV, or air conditioner.

Figure 8-1. IR Controlled Home Appliances



- Toys: Various toys for kids such as cars, trucks, helicopters, or robots using IR remotes for control.

Figure 8-2. Remote Controlled Toy Car



9. Conclusion

In this application note, an overview of capacitive touch sensing implementation using Microchip's ADC² with hardware Capacitive Voltage Divider (CVD) technique, and mTouch capacitive sensing library module for MCC is discussed.

This document also provides an overview of IR communication with NEC protocol. The IR transmitter and touch interface are implemented using ADC² with hardware CVD, CLCs, PWM and DSM peripherals of the PIC18 Q10 microcontroller. The IR receiver and DC motor control are implemented using CCP and PWM peripherals of the PIC18 Q10 microcontroller.

The application note aims to highlight the important features of the PIC18 Q10 MCUs for real-time control applications. The PIC18 Q10 family of MCUs are featured with powerful Core Independent Peripherals, along with the other digital and advanced analog peripherals. Hence, using these MCUs for the application development reduces the system design cost and makes the system more power conservative, reliable and deterministic. These MCUs can be used for a wide range of general purpose, low-power, reliable, and deterministic real-time control applications such as remote control of various home appliances, remote controlled toys for kids, and many other applications.

10. References

- mTouch[®] Capacitive Sensing Library Module for MPLAB[®] X Code Configurator User's Guide: <http://ww1.microchip.com/downloads/en/DeviceDoc/40001852A.pdf>
- AN1478: mTouch[®] Sensing Solution Acquisition Methods Capacitive Voltage Divider
- NEC IR Protocol: <https://techdocs.altium.com/display/FPGA/NEC+Infrared+Transmission+Protocol>
- IR Click: <https://www.mikroe.com/ir-click>
- DC Motor 8 Click: <https://www.mikroe.com/dc-motor-8-click>

11. Appendix

11.1 Advantages and Challenges of IR Communication for Real-Time Control Applications

IR communication is used in data communication for monitoring and control applications having a short range, such as 10-30 meters. Following are the advantages of using IR communication:

- Inexpensive, compared to other technologies.
- Works over a moderate bandwidth of 115 kbps.
- Works well over a short distance.

There are few disadvantages of infrared communications:

- Infrared frequencies are affected by hard objects, for example, walls, doors, smoke, dust, fog, or sunlight. Hence, it does not work through walls or doors. It requires a line of sight between transmitter and receiver to communicate.
- It can control only one device at one time.
- It has short range of a few meters. Its performance degrades with longer distances.
- Infrared waves at high power can damage eyes.
- It supports lower data rate transmission compare to wired transmission.

The Microchip Web Site

Microchip provides online support via our web site at <http://www.microchip.com/>. This web site is used as a means to make files and information easily available to customers. Accessible by using your favorite Internet browser, the web site contains the following information:

- **Product Support** – Data sheets and errata, application notes and sample programs, design resources, user's guides and hardware support documents, latest software releases and archived software
- **General Technical Support** – Frequently Asked Questions (FAQ), technical support requests, online discussion groups, Microchip consultant program member listing
- **Business of Microchip** – Product selector and ordering guides, latest Microchip press releases, listing of seminars and events, listings of Microchip sales offices, distributors and factory representatives

Customer Change Notification Service

Microchip's customer notification service helps keep customers current on Microchip products. Subscribers will receive e-mail notification whenever there are changes, updates, revisions or errata related to a specified product family or development tool of interest.

To register, access the Microchip web site at <http://www.microchip.com/>. Under "Support", click on "Customer Change Notification" and follow the registration instructions.

Customer Support

Users of Microchip products can receive assistance through several channels:

- Distributor or Representative
- Local Sales Office
- Field Application Engineer (FAE)
- Technical Support

Customers should contact their distributor, representative or Field Application Engineer (FAE) for support. Local sales offices are also available to help customers. A listing of sales offices and locations is included in the back of this document.

Technical support is available through the web site at: <http://www.microchip.com/support>

Microchip Devices Code Protection Feature

Note the following details of the code protection feature on Microchip devices:

- Microchip products meet the specification contained in their particular Microchip Data Sheet.
- Microchip believes that its family of products is one of the most secure families of its kind on the market today, when used in the intended manner and under normal conditions.
- There are dishonest and possibly illegal methods used to breach the code protection feature. All of these methods, to our knowledge, require using the Microchip products in a manner outside the operating specifications contained in Microchip's Data Sheets. Most likely, the person doing so is engaged in theft of intellectual property.
- Microchip is willing to work with the customer who is concerned about the integrity of their code.

- Neither Microchip nor any other semiconductor manufacturer can guarantee the security of their code. Code protection does not mean that we are guaranteeing the product as “unbreakable.”

Code protection is constantly evolving. We at Microchip are committed to continuously improving the code protection features of our products. Attempts to break Microchip’s code protection feature may be a violation of the Digital Millennium Copyright Act. If such acts allow unauthorized access to your software or other copyrighted work, you may have a right to sue for relief under that Act.

Legal Notice

Information contained in this publication regarding device applications and the like is provided only for your convenience and may be superseded by updates. It is your responsibility to ensure that your application meets with your specifications. MICROCHIP MAKES NO REPRESENTATIONS OR WARRANTIES OF ANY KIND WHETHER EXPRESS OR IMPLIED, WRITTEN OR ORAL, STATUTORY OR OTHERWISE, RELATED TO THE INFORMATION, INCLUDING BUT NOT LIMITED TO ITS CONDITION, QUALITY, PERFORMANCE, MERCHANTABILITY OR FITNESS FOR PURPOSE. Microchip disclaims all liability arising from this information and its use. Use of Microchip devices in life support and/or safety applications is entirely at the buyer’s risk, and the buyer agrees to defend, indemnify and hold harmless Microchip from any and all damages, claims, suits, or expenses resulting from such use. No licenses are conveyed, implicitly or otherwise, under any Microchip intellectual property rights unless otherwise stated.

Trademarks

The Microchip name and logo, the Microchip logo, AnyRate, AVR, AVR logo, AVR Freaks, BitCloud, chipKIT, chipKIT logo, CryptoMemory, CryptoRF, dsPIC, FlashFlex, flexPWR, Helder, JukeBlox, KeeLoq, Klear, LANCheck, LINK MD, maXStylus, maXTouch, MediaLB, megaAVR, MOST, MOST logo, MPLAB, OptoLyzer, PIC, picoPower, PICSTART, PIC32 logo, Prochip Designer, QTouch, SAM-BA, SpyNIC, SST, SST Logo, SuperFlash, tinyAVR, UNI/O, and XMEGA are registered trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

ClockWorks, The Embedded Control Solutions Company, EtherSynch, Hyper Speed Control, HyperLight Load, IntelliMOS, mTouch, Precision Edge, and Quiet-Wire are registered trademarks of Microchip Technology Incorporated in the U.S.A.

Adjacent Key Suppression, AKS, Analog-for-the-Digital Age, Any Capacitor, AnyIn, AnyOut, BodyCom, CodeGuard, CryptoAuthentication, CryptoAutomotive, CryptoCompanion, CryptoController, dsPICDEM, dsPICDEM.net, Dynamic Average Matching, DAM, ECAN, EtherGREEN, In-Circuit Serial Programming, ICSP, INICnet, Inter-Chip Connectivity, JitterBlocker, KlearNet, KlearNet logo, memBrain, Mindi, MiWi, motorBench, MPASM, MPF, MPLAB Certified logo, MPLIB, MPLINK, MultiTRAK, NetDetach, Omniscient Code Generation, PICDEM, PICDEM.net, PICKit, PICtail, PowerSmart, PureSilicon, QMatrix, REAL ICE, Ripple Blocker, SAM-ICE, Serial Quad I/O, SMART-I.S., SQI, SuperSwitcher, SuperSwitcher II, Total Endurance, TSHARC, USBCheck, VariSense, ViewSpan, WiperLock, Wireless DNA, and ZENA are trademarks of Microchip Technology Incorporated in the U.S.A. and other countries.

SQTP is a service mark of Microchip Technology Incorporated in the U.S.A.

Silicon Storage Technology is a registered trademark of Microchip Technology Inc. in other countries.

GestIC is a registered trademark of Microchip Technology Germany II GmbH & Co. KG, a subsidiary of Microchip Technology Inc., in other countries.

All other trademarks mentioned herein are property of their respective companies.

© 2019, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.

ISBN: 978-1-5224-4443-5

Quality Management System Certified by DNV

ISO/TS 16949

Microchip received ISO/TS-16949:2009 certification for its worldwide headquarters, design and wafer fabrication facilities in Chandler and Tempe, Arizona; Gresham, Oregon and design centers in California and India. The Company's quality system processes and procedures are for its PIC[®] MCUs and dsPIC[®] DSCs, KEELOQ[®] code hopping devices, Serial EEPROMs, microperipherals, nonvolatile memory and analog products. In addition, Microchip's quality system for the design and manufacture of development systems is ISO 9001:2000 certified.

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://www.microchip.com/support Web Address: www.microchip.com	Australia - Sydney Tel: 61-2-9868-6733 China - Beijing Tel: 86-10-8569-7000 China - Chengdu Tel: 86-28-8665-5511 China - Chongqing Tel: 86-23-8980-9588 China - Dongguan Tel: 86-769-8702-9880 China - Guangzhou Tel: 86-20-8755-8029 China - Hangzhou Tel: 86-571-8792-8115 China - Hong Kong SAR Tel: 852-2943-5100 China - Nanjing Tel: 86-25-8473-2460 China - Qingdao Tel: 86-532-8502-7355 China - Shanghai Tel: 86-21-3326-8000 China - Shenyang Tel: 86-24-2334-2829 China - Shenzhen Tel: 86-755-8864-2200 China - Suzhou Tel: 86-186-6233-1526 China - Wuhan Tel: 86-27-5980-5300 China - Xian Tel: 86-29-8833-7252 China - Xiamen Tel: 86-592-2388138 China - Zhuhai Tel: 86-756-3210040	India - Bangalore Tel: 91-80-3090-4444 India - New Delhi Tel: 91-11-4160-8631 India - Pune Tel: 91-20-4121-0141 Japan - Osaka Tel: 81-6-6152-7160 Japan - Tokyo Tel: 81-3-6880-3770 Korea - Daegu Tel: 82-53-744-4301 Korea - Seoul Tel: 82-2-554-7200 Malaysia - Kuala Lumpur Tel: 60-3-7651-7906 Malaysia - Penang Tel: 60-4-227-8870 Philippines - Manila Tel: 63-2-634-9065 Singapore Tel: 65-6334-8870 Taiwan - Hsin Chu Tel: 886-3-577-8366 Taiwan - Kaohsiung Tel: 886-7-213-7830 Taiwan - Taipei Tel: 886-2-2508-8600 Thailand - Bangkok Tel: 66-2-694-1351 Vietnam - Ho Chi Minh Tel: 84-28-5448-2100	Austria - Wels Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 Denmark - Copenhagen Tel: 45-4450-2828 Fax: 45-4485-2829 Finland - Espoo Tel: 358-9-4520-820 France - Paris Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 Germany - Garching Tel: 49-8931-9700 Germany - Haan Tel: 49-2129-3766400 Germany - Heilbronn Tel: 49-7131-67-3636 Germany - Karlsruhe Tel: 49-721-625370 Germany - Munich Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 Germany - Rosenheim Tel: 49-8031-354-560 Israel - Ra'anana Tel: 972-9-744-7705 Italy - Milan Tel: 39-0331-742611 Fax: 39-0331-466781 Italy - Padova Tel: 39-049-7625286 Netherlands - Drunen Tel: 31-416-690399 Fax: 31-416-690340 Norway - Trondheim Tel: 47-72884388 Poland - Warsaw Tel: 48-22-3325737 Romania - Bucharest Tel: 40-21-407-87-50 Spain - Madrid Tel: 34-91-708-08-90 Fax: 34-91-708-08-91 Sweden - Gothenberg Tel: 46-31-704-60-40 Sweden - Stockholm Tel: 46-8-5090-4654 UK - Wokingham Tel: 44-118-921-5800 Fax: 44-118-921-5820