MICROCHIP

AN2912

Using CLCs in Real-Time Applications

Introduction

Authors: Ankit Tripathi,
Srinivasa K R, Microchip Technology Inc

The Configurable Logic Cell (CLC) is a flexible peripheral that enables creation of on-chip custom logic functions for PIC® microcontrollers. This peripheral allows the user to specify combinations of signals as inputs to a logic function, and can route the logic output to other peripherals and I/O pins. It can be configured with basic logic functions such as logic gates, flip-flops and latches.

The CLC is a Core Independent Peripheral (CIP), which means it performs its tasks with no code or supervision from the CPU after configuration is completed. It operates independently and outside the speed limitations of the PIC microcontroller's CPU. This not only decreases the system's response time, it also reduces the software development time, as some of the functions typically done in code are instead done in hardware.

The CLC can be used to build custom logic functions, and also to internally connect other peripherals like timers, PWMs, serial ports and I/O pins, allowing hardware customization with unprecedented ease.

Scope

This application note introduces the CLC in PIC® microcontrollers and presents two examples of how it may be used to simplify the design of a real-world system. This application note is supplemented by the respective firmware of the two case studies. To access the firmware visit http://www.microchip.com/wwwproducts/en/PIC18F47Q10. The following content is discussed in this application note:

- Overview of CLC in PIC Devices
- Features and Benefits of CLC
- CLC in Real-Time Applications
- Finite State Machine Implementation using CLCs
- Encoder and Decoder Implementation using CLCs

Table of Contents

Introduction......1

Scope......1

1. Overview......3
   1.1. Features......3
   1.2. Benefits......4

2. CLC in the PIC18 Q10 Family......5

3. CLC in Real-Time Applications......6

4. Application 1: Finite State Machine......7
   4.1. Sequential Logic Circuits......7
   4.2. Case Study - Vending Machine......7
   4.3. Finite State Machine Demonstration......23

5. Application 2: Encoders and Decoders......24
   5.1. Encoder......24
   5.2. Decoder......24
   5.3. Case Study – Gray Code Decoder......24
   5.4. Gray Code Decoder Demonstration......35

6. Conclusion......36

7. References......37

The Microchip Web Site......38

Customer Change Notification Service......38

Customer Support......38

Microchip Devices Code Protection Feature......38

Legal Notice......39

Trademarks......39

Quality Management System Certified by DNV......40

Worldwide Sales and Service......41

1. Overview

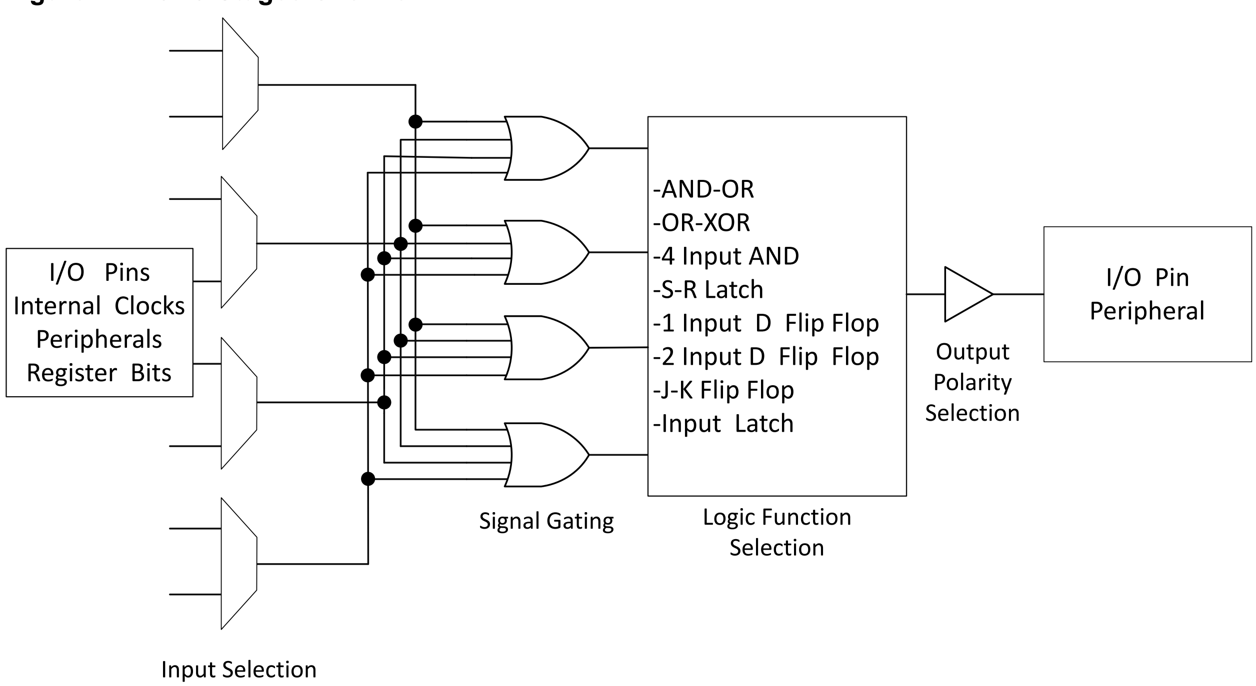
The CLC is a user-configurable peripheral, similar to a Programmable Logic Device (PLD). Various internal and external signals can be used as inputs to the CLC. These signals from other peripherals or from input pins, are routed to the CLC's specified combinatorial and sequential logic. CLC performs the intended logic operation and provides an output that can be routed to other peripherals, or can be sent to an I/O pin for external circuit management.

1.1 Features

The CLC in PIC devices has a range of input and output selection possibilities. In general, configuring the CLC involves four steps. A brief insight into the four stages of the CLC peripheral is as follows:

- **Input Selection**
  The CLC can receive various signals, such as input pins, internal clocks, outputs of other peripherals, and events of peripherals, such as a timer input.
- **Signal Gating**
  The selected input signal sources can be directed to the desired logic function through the signal gating stage.
- **Logic Function Selection**
  In the CLC, the outputs of the data gating stage are inputs to the logic function selection stage. The CLC supports logic functions such as AND-OR, OR-XOR, AND, SR latch, D flip-flops and J-K flip-flops.
- **Output Polarity Selection**
  The output polarity stage is used to select the desired polarity of the logic output.

Figure 1-1. CLC Stages Overview



For more details refer to the device-specific data sheet.

1.2 Benefits

In addition to implementing common logic functions, the CLC can be used to create custom logic operations. Individual CLC interrupt triggers allow combinations of many possible input/output signals in real-time control systems. Some of the advantages of CLC usage include:

- The CLC can be used as a stand-alone peripheral in implementing sequential and/or combinational logic functions, thus facilitating quick event triggers and responses.
- The CLC supports a higher level of integration without any external components, thus reducing PCB size and system costs.
- Logic functions implemented in hardware have real-time event response when compared to logic functions implemented in software.
- The CLC, used in conjunction with other peripherals, extends the capabilities of those peripherals by allowing custom complex function implementation in the hardware.
- Since the CLC is completely core independent, it effectively reduces the CPU bandwidth requirement for any application, as many simple logic and event responses can be offloaded from the CPU to the peripheral.
- The CLC reduces Flash and RAM requirements as software algorithms are not required.
- Being a true CIP, the CLC can be used in Power-Down modes for any logical or sequential operation, or for creating CPU wake-up conditions.

2. CLC in the PIC18 Q10 Family

Devices in the PIC18 Q10 family are equipped with eight Configurable Logic Cells (CLCs), which provide programmable logic that operates outside the speed limitations of software execution. Each logic cell takes up to 64 input signals and, using configurable gates, reduces the 64 inputs to four logic lines that drive one of eight selectable single output logic functions.

The CLC output can be steered to the inputs of many of the peripherals or external pins to support design requirements. Also, The CLC output can generate configurable rising-edge or falling-edge interrupts, or both if software intervention or quick decision making is required in the application. These interrupts can also be used to create complex conditions for MCU wake-up.

The CLC module can be configured as combinational logic or as latches. Possible configurations are:

- **Combinational logic:**
  - AND
  - NAND
  - AND-OR
  - OR-XOR
- **Latches:**
  - SR
  - Clocked D with Set and Reset
  - Transparent D with Set and Reset
  - Clocked J-K with Reset

3. CLC in Real-Time Applications

The versatility and simplicity of the CLC extend the capabilities of a PIC microcontroller for real-time applications. Short duration application signals can be easily missed because of the asynchronous polling. However, the CLC can be used to store the transition on the input, which the CPU can check at later time. It is next to impossible for firmware to monitor two signals in real time and take output all at the same time, as at least one clock cycle delay would be required to execute these operations. Instead, the CLC can be used for actual real-time operations.

Real-Time Application Examples:

- Boolean Expressions
- Sequential Circuits
- Encoders and Decoders
- Counters
- Multiplexers
- Real-Time Data Capturing without Interrupt
- Complex Waveform Generation (e.g., NEC Code for IR transmission)
- Cyclo-Converters
- Phase Detectors
- Frequency Divider
- Pulse Extender
- PWM Steering
- Signal Threshold and Hold Circuit
- Switch Debouncing

4. Application 1: Finite State Machine

A finite state machine (FSM) is a logical unit or device that stores the state of operation at a given time and can perform transition of state in response to any change in external inputs. A FSM is defined by a list of its states, its initial state, and the conditions for each transition.

To summarize, a state machine can be described in four parts:

- An initial state or record of something stored someplace
- A set of possible input events
- A set of new states that may result from the input changes
- A set of possible actions or output events that result from a new state

4.1 Sequential Logic Circuits

Sequential logic is a type of logic circuit whose output depends not only on the present value of its input signals but also on the sequence of past inputs and input history. The past values of the inputs are held in logic storage devices. All circuits in practical digital devices are a mixture of combinational and sequential logic.

4.2 Case Study - Vending Machine

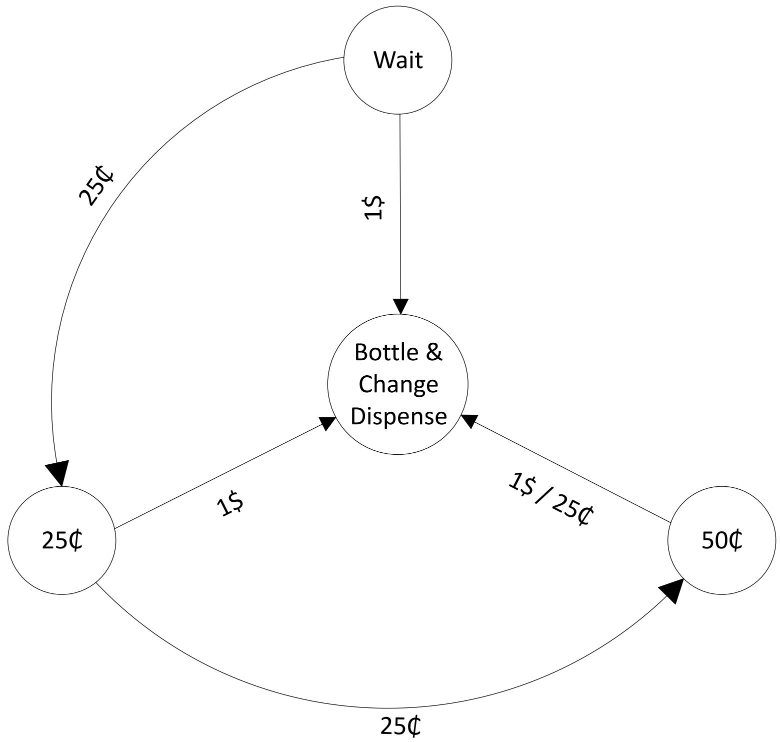
The vending machine is one of the simple yet thorough examples of a finite state machine.
Consider the vending machine selling water or drink in a bottle or can for 75 cents.

- Customers can insert either a dollar or quarters (25 cents). Once a sufficient amount of money is inserted, the vending machine dispenses a bottle/can.
  - If the customer inserts a dollar, it will return one quarter as change along with the purchased bottle/can.
  - If the customer inserts three quarters a bottle/can will be dispensed.
- A “*Money Receiver*” detects when money has been inserted in the vending machine.

  The receiver sends two logic signals to the circuit indicating whether a dollar or quarter was inserted.
- A “*Bottle/Can Dispenser*” system holds the bottles and will release one bottle when its input signal is asserted.
- A “*Coin Return*” system holds quarters for change and will release coins when its input signal is asserted.
- Depending on the inserted amount of money, it returns the correct change, as follows:
  - It returns a quarter if a dollar is inserted.
  - It returns two quarters if a dollar is inserted after a quarter.
  - It returns three quarters if a dollar is inserted after two quarters.

  In all of the above cases, the bottle/can dispenser releases a bottle or a can.
- The system also generates an error condition when a dollar and a quarter are inserted simultaneously. In case of error all the coins are returned back without bottle dispense.

Figure 4-1. Vending Machine State Diagram



4.2.1 Vending Machine - State Transition Table and Logic Equations

Every state diagram can be interpreted in the form of a transition table. Breaking up the vending machine example results in a state transition table as shown in Table 4-1.

Table 4-1. State Transition Table for Vending Machine

Input					Output					
Current State			Dollar	Quarter	Next State		Bottle	Quarter		Error
State	A	B	C	D	P	Q	R	S	T	E
Wait	0	0	0	0	0	0	0	0	0	0
Wait	0	0	0	1	0	1	0	0	0	0
Wait	0	0	1	0	0	0	1	0	1	0
Wait	0	0	1	1	0	0	0	0	0	1
25₵	0	1	0	0	0	1	0	0	0	0
25₵	0	1	0	1	1	0	0	0	0	0
25₵	0	1	1	0	0	0	1	1	0	0
25₵	0	1	1	1	0	0	0	0	0	1
50₵	1	0	0	0	1	0	0	0	0	0
50₵	1	0	0	1	0	0	1	0	0	0
50₵	1	0	1	0	0	0	1	1	1	0
50₵	1	0	1	1	0	0	0	0	0	1

This Boolean table can be easily simplified using the K-map technique into individual expressions for each of the outputs, giving the result in the form of simple logical equations.

Solving for next state:

- P = B C'D + A C'D'
- Q = A'B'C'D + B C'D'

Solving for dispense outputs and error state:

- R = C D' + A C'D
- S = B C D' + A C D'
- T = B'C D'
- E = C D

Apart from these logical operations it can be observed that the previous state is also needed for computing the next states and dispense signals. Hence, sequential circuit using D flip-flop based latch can be used to save previous state.

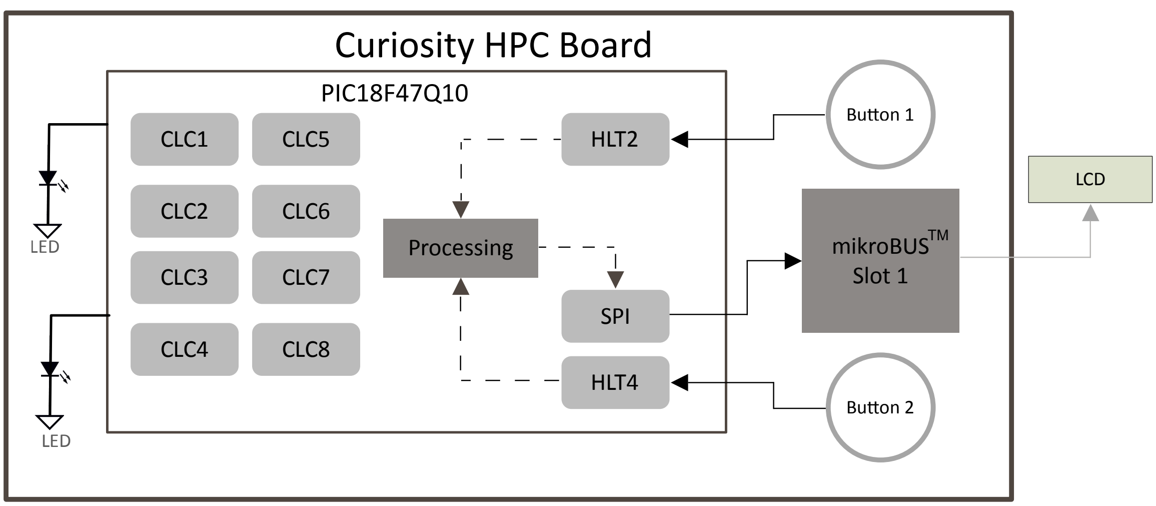
- A = previous of (P)
- B = previous of (Q)

4.2.2 Vending Machine Example using Curiosity HPC

Curiosity HPC is a fully-integrated 8-bit development platform designed from the ground-up to take full advantage of Microchip's MPLAB® X development environment.

Curiosity HPC includes an integrated programmer/debugger, and requires no additional hardware to get started. It supports 28- or 40-pin 8-bit PIC® microcontrollers with low-voltage programming capability. With two buttons, four LEDs and two mikroBUS™ connectors on the board, it provides the perfect platform for demonstrating the vending machine example.

Figure 4-2. Vending Machine using Curiosity HPC Development Board



4.2.3 Application Overview

The general idea of the application is to implement all the logical computations using CLCs. All the logical expressions derived from section 4.2.1 Vending Machine - State Transition Table and Logic Equations can be performed using CLCs. By setting them in various gate configurations, the CLCs are used to get the required logical expression.

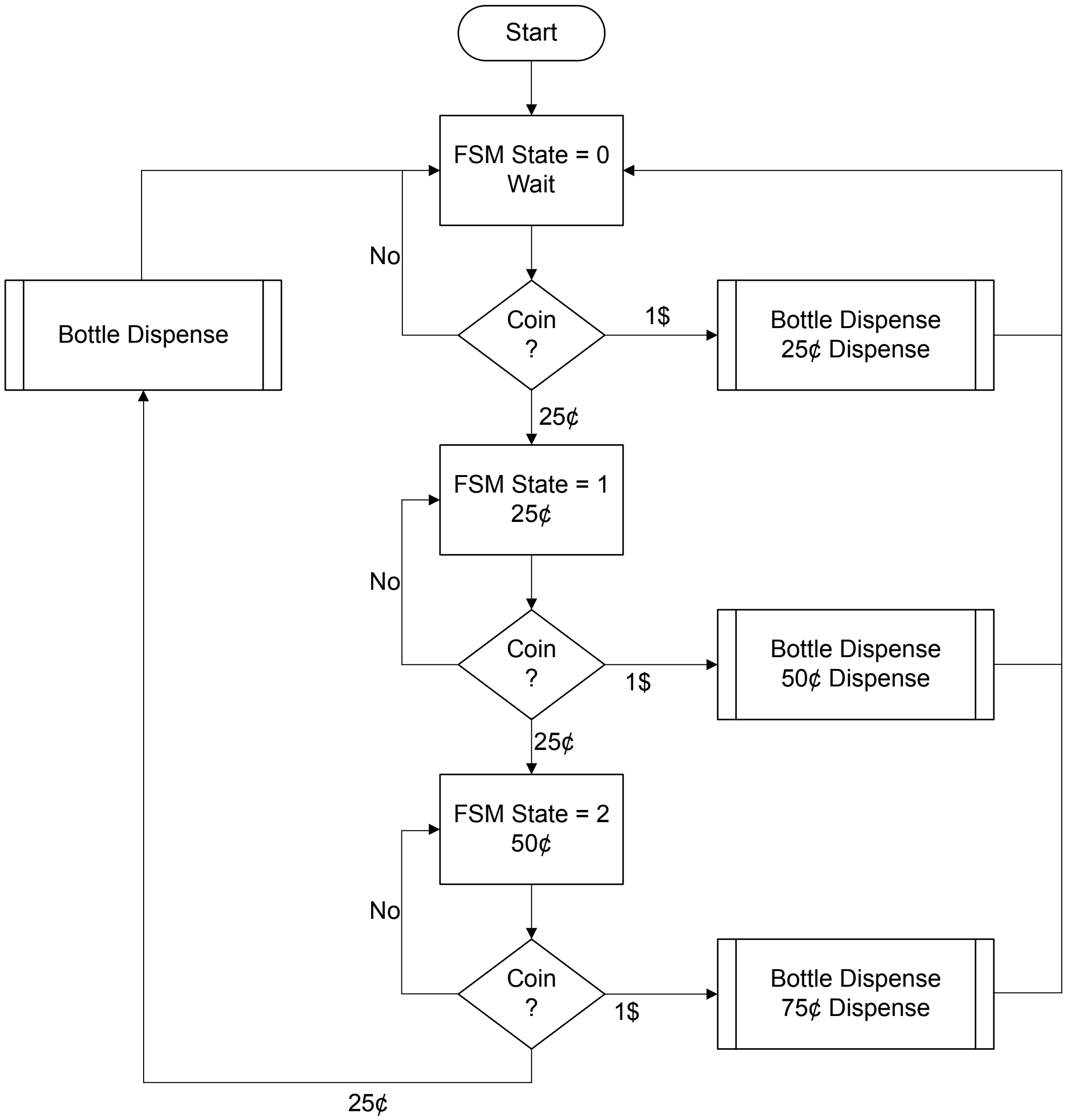
The required inputs are *dollar*, *quarter* and *current state*, whereas outputs are *next state*, *dispense of bottle* and *change*. To implement the vending machine using the Curiosity HPC board, some simulation is required to create full application environment. Hence, for demonstration, the *dollar* and *quarter* inputs are simulated using push buttons *S2* and *S1*, respectively, present on the board.

For switch debouncing, timers TMR2 and TMR4, in Monostable mode, are used for each switch. These timers will generate one low pulse of programmed time at each switch press. As this pulse is of low polarity, each switch press will give logic '`0`', so it needs to be inverted before input to CLC. This can be easily implemented by using the input polarity selection of the CLC.

Similarly, for bottle and change dispense output indications on-board LEDs are used. LED D5 is used to show bottle dispense whereas LEDs D4, D3 and D2 are used to show each quarter out. Depending on the money received, up to three quarters can be released. The output is triggered by CLC interrupts of the respective CLC output, where the LEDs are turned on. The ON time of the LEDs is controlled by a dedicated timer, TMR5. When the LEDs are turned ON, the timer module is also started, which will generate an interrupt after three seconds. This interrupt is used to turn OFF the LEDs and stop the TMR5 again.

FSM states and dispense messages are displayed accordingly on the LCD, in synchronization with the system state and dispense outputs. The LCD mini click from MikroElektronika is used as display unit, which is supported by the MPLAB® Code Configurator (MCC). MCC generates all the required driver files for LCD and the display routines are implemented at the application level. After some period of inactivity, set by the user, the controller is put into IDLE mode and the LCD backlight is turned off, to reduce power consumption. The system automatically wakes up on any switch-press event, which is implemented by using the switches' timer output interrupt. The same interrupt is used to clear the user inactivity time-out. Figure 4-3 provides the overview of the vending machine application flow diagram.

Figure 4-3. Vending Machine Application Flow Diagram



4.2.4 Hardware Setup

The Curiosity HPC (Figure 4-4) with a PIC18F47Q10 microcontroller is used for demonstrating the vending machine example. The vending machine demo also uses an optional LCD display for displaying the FSM states and dispense messages, so a LCD mini click (MIKROE-2453) from MikroElektronika is used (Figure 4-5). The LCD click is used with mikroBUS slot 1.

Figure 4-4. Curiosity HPC Board

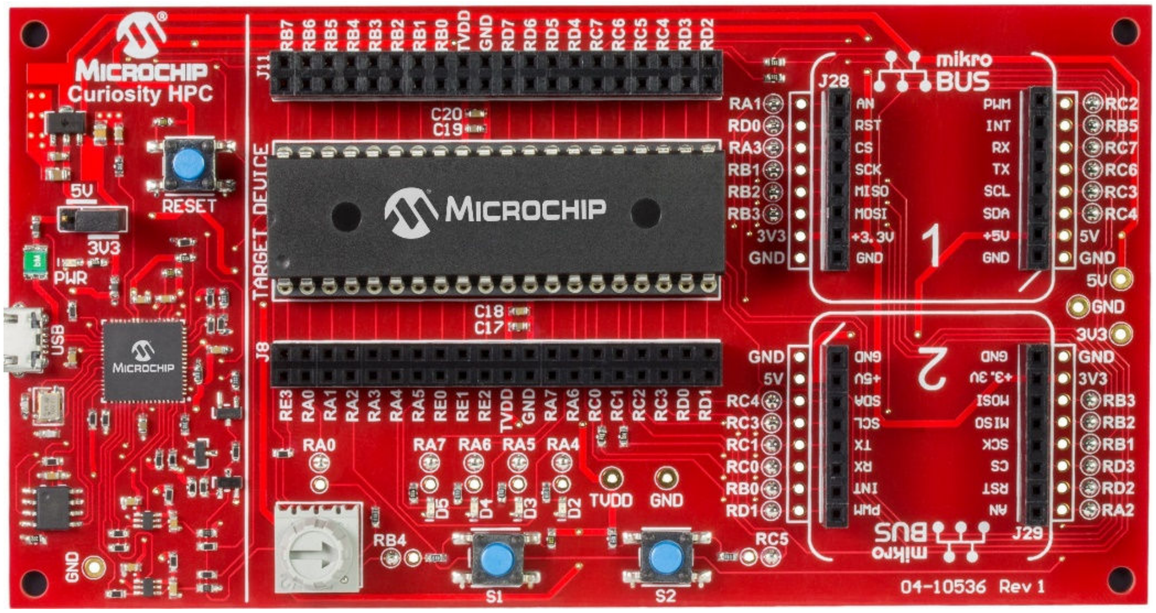


Figure 4-5. LCD Mini Click

4.2.5 Pin Mapping

As discussed in the 4.2.3 Application Overview section, the demo requires two switches for simulating the coin inserts of dollars and quarters, four LEDs for simulating dispense outputs of bottle/can and change coins, and one LCD display. Table 4-2 shows the pin configuration of PIC18 Q10 MCU and the module tied to it for all the required peripherals.

Table 4-2. Pin Details for Vending Machine

Pin Function	Pin Name	Pin Type	Used As
LCD CS	RA3	Output-Digital	LCD I/O
LCD CS2	RA1	Output-Digital	LCD I/O
LCD Reset	RD0	Output-Digital	LCD I/O
LCD BL	RC2	Output-Digital	LCD I/O
LCD SDI	RB2	Input-Digital	LCD SPI
LCD SDO	RB3	Output-Digital	LCD SPI
LCD SCK	RB1	Output-Digital	LCD SPI
Bottle LED	RA7	Output-Digital	I/O
Change 1 LED	RA4	Output-Digital	I/O
Change 2 LED	RA5	Output-Digital	I/O
Change 3 LED	RA6	Output-Digital	I/O
Dollar Switch	RB4	Input-Digital	TMR4 IN
Quarter Switch	RC5	Input-Digital	TMR2 IN

4.2.6 Software Tools

Microchip's MPLAB X IDE, compiler and graphical code generators are used throughout the application firmware development to provide an easy and hassle-free user experience. Following are the tools used for this demo application:

- MPLAB X IDE v5.20
- XC8 Compiler v2.05
- MPLAB Code Configurator (MCC) v3.75

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

4.2.7 MCC Configuration

MPLAB® Code Configurator (MCC) is a graphical programming environment that generates seamless, easy to understand C code to give a head start to the project, saving the designer's time to initialize and configure all the modules, and to go through the data sheets. Using an instructive interface, it enables and configures all peripherals and functions specific to the application requirements. For more details visit http://www.microchip.com/mplab/mplab-code-configurator.

MCC is used for this demo to generate the initialization and peripheral configuration code:

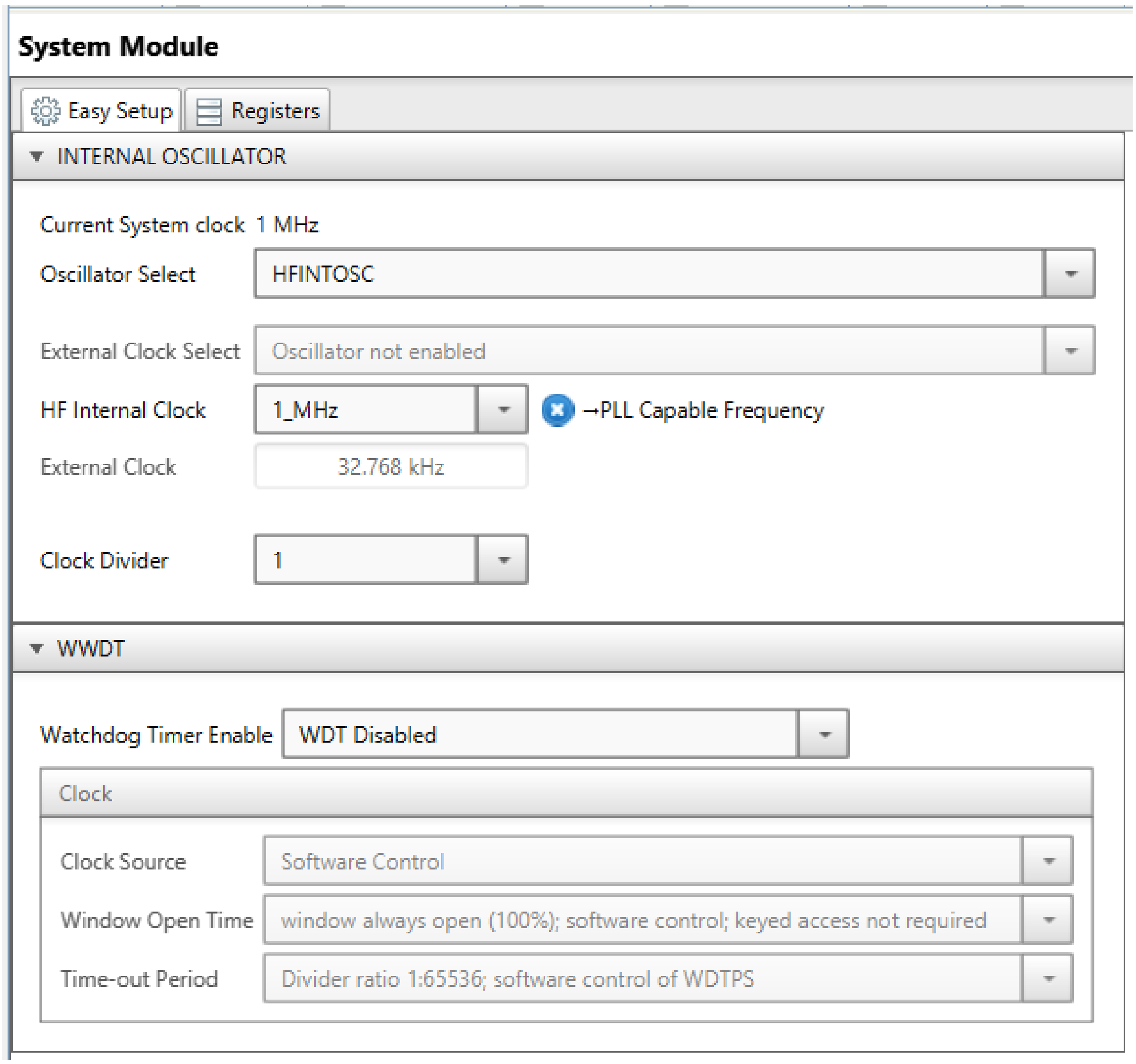
- System configuration for clock settings
- CLC configurations for logic expressions
- Timer configurations for switch debouncing and LED on-time delay
- LCD mini click SPI configuration and respective code generation

These configuration steps are detailed below:

- **System Configuration**

  The system clock is taken as 1 MHz and is generated from HFINTOSC. Watchdog Timer is disabled.

Figure 4-6. System Configuration via MCC



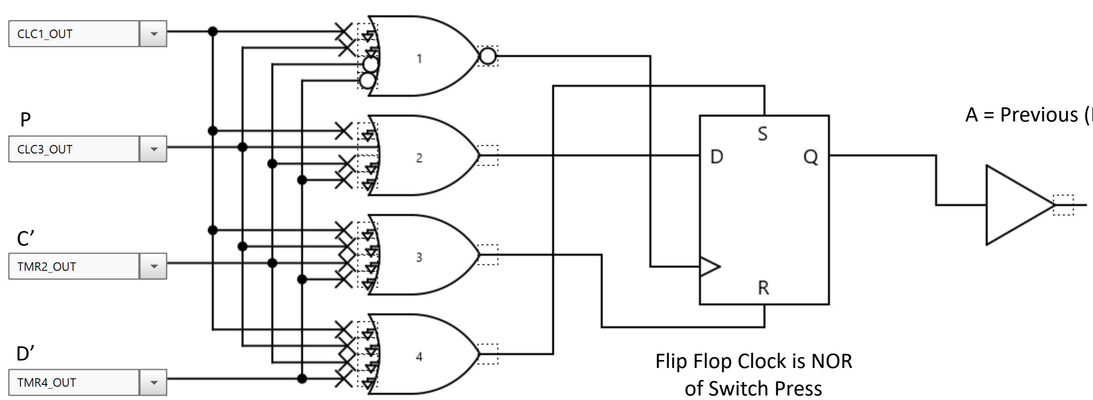
- **CLC Configuration**

  The CLCs are configured to get the logical expressions for each variable and to generate wake-up signal for the MCU. The equations described in 4.2.1 Vending Machine - State Transition Table and Logic Equations are logically configured using the CLC. Table 4-3 lists the equations with some rearrangements for AND of OR (product of sum) format as to be implemented via CLC.

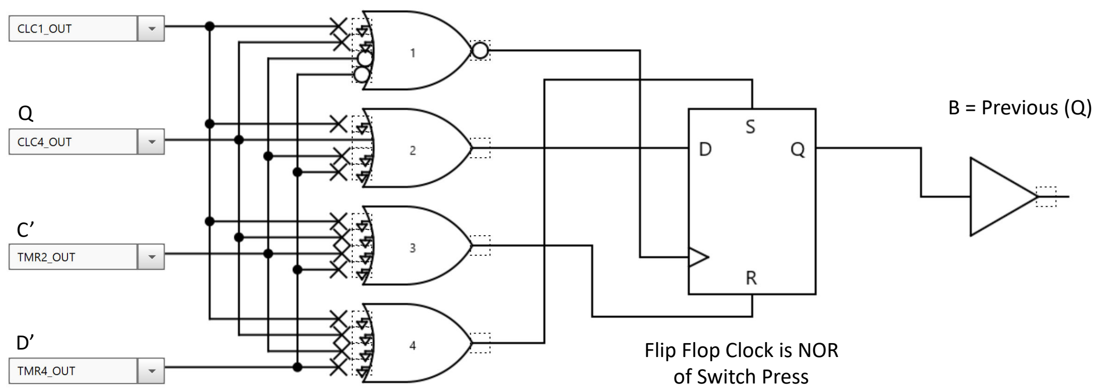
Table 4-3. CLC Settings

Sr. No.	Reference	Logical Expression	Function	Peripheral
1	A	Previous (P)	Current State MSB	CLC 1
2	B	Previous (Q)	Current State LSB	CLC 2
3	C	Inverse of Timer 2 output	Dollar in	TMR 2
4	D	Inverse of Timer 4 output	Quarter in	TMR 4
5	P	[(B’ + C + D’) (A’ + C + D)]’	Next State MSB	CLC 3
6	Q	[(A + B + C + D’) (B’ + C + D)]’	Next State LSB	CLC 4
7	R	[(C’ + D) (A’ + C + D’)]’	Bottle out	CLC 5
8	S	[(B’ + C’ + D) (A’ + C’ + D)]’	Change out MSB	CLC 6
9	T	B’ C D’	Change out LSB	CLC 7
10	E	C D	Error State	CLC 8

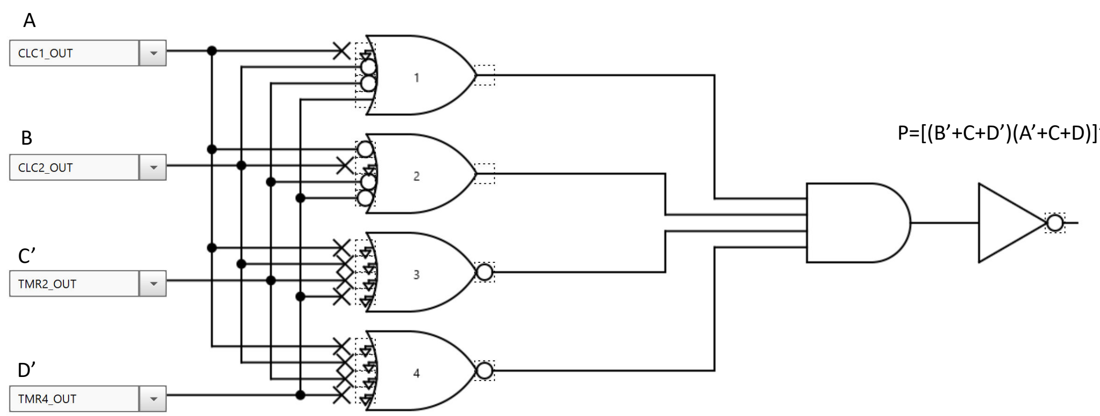
1. CLC1: 1-input D flip-flop



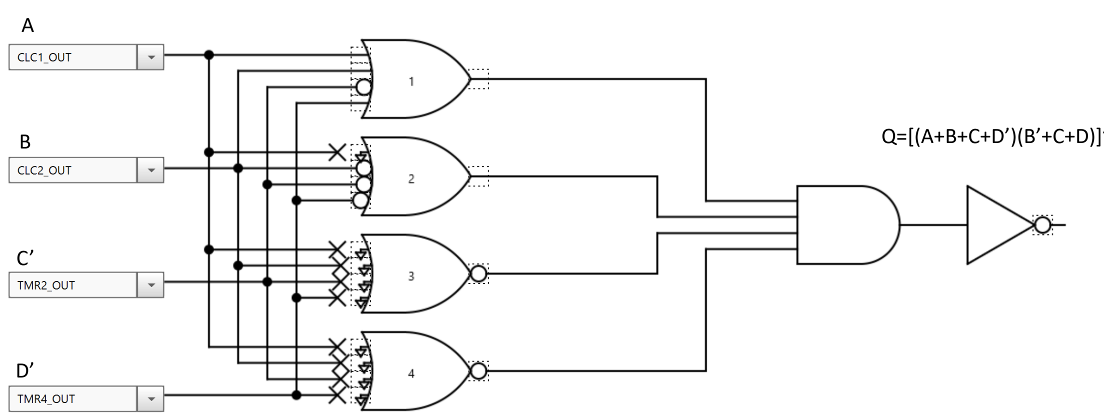
2. CLC2: 1-input D flip-flop



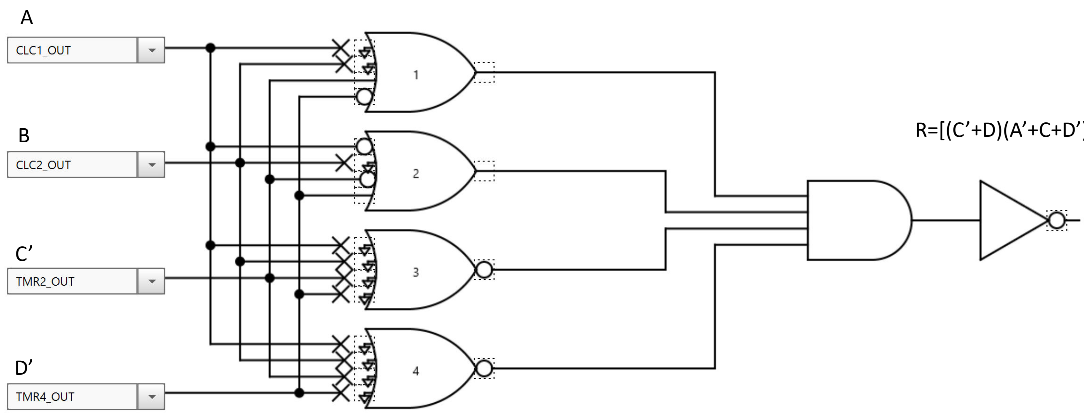
3. CLC3: 4-input AND



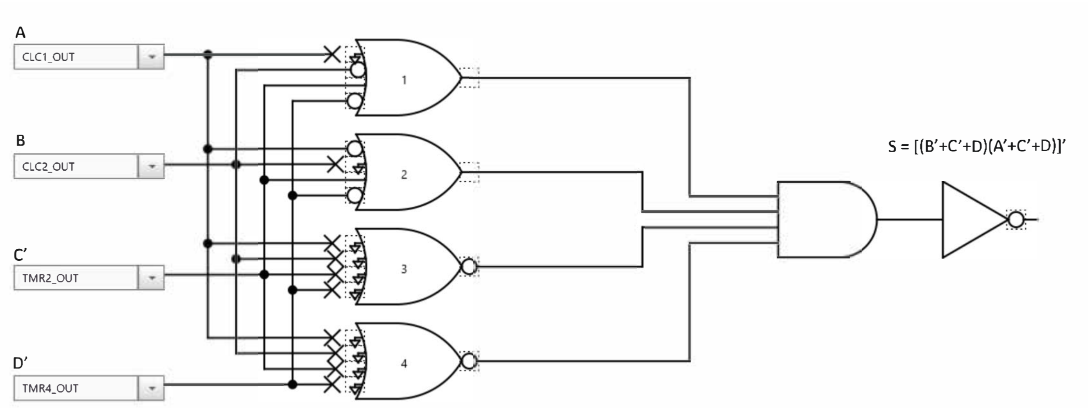
4. CLC4: 4-input NAND



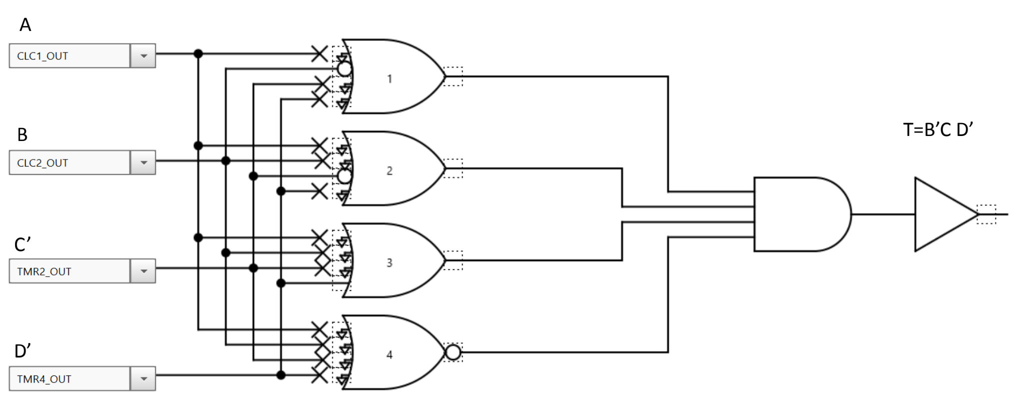
5. CLC5: 4-input AND, CLC Interrupt Enable (Rising Edge)



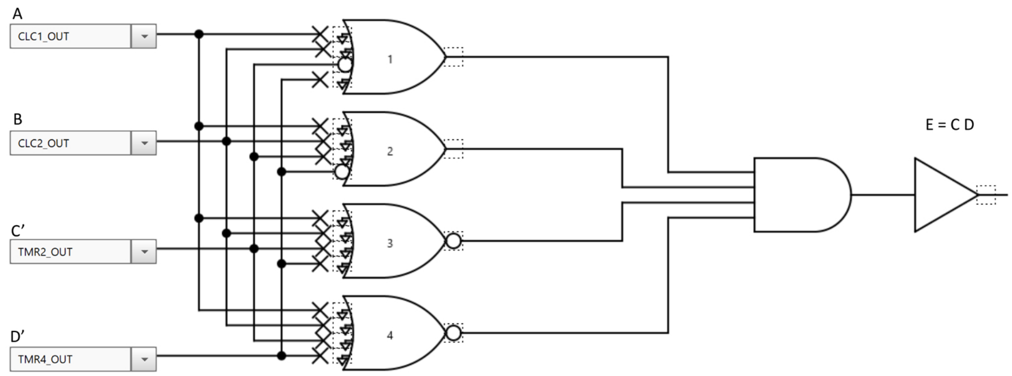
6. CLC6: 4-input AND, CLC Interrupt Enable (Rising Edge)



7. CLC7: 4-input AND, CLC Interrupt Enable (Rising Edge)



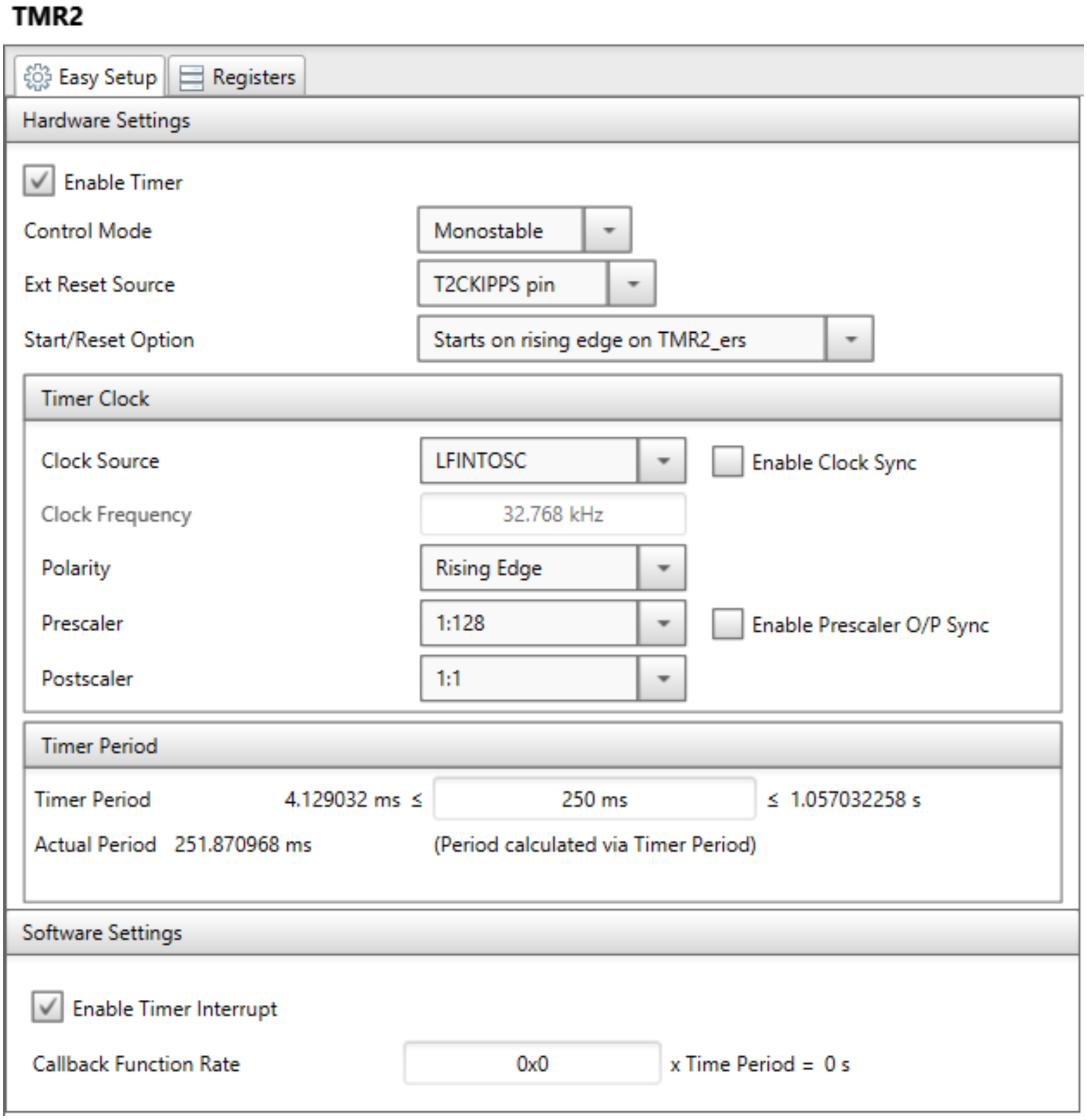
8. CLC8: 4-input AND, CLC Interrupt Enable (Rising Edge)



- **Timer Configuration**

  As discussed in 4.2.3 Application Overview, TMR2 and TMR4 are used in Monostable mode for switch debouncing, a period of 250 ms is programmed for the debounce. The timer is triggered with the rising edge of the switch release event. The timer clock is set to get the desired timer period only, and need not be identical to the settings in this application. The timer output is directly fed into the CLC, yet timer interrupts are enabled for wake-up from Sleep mode.

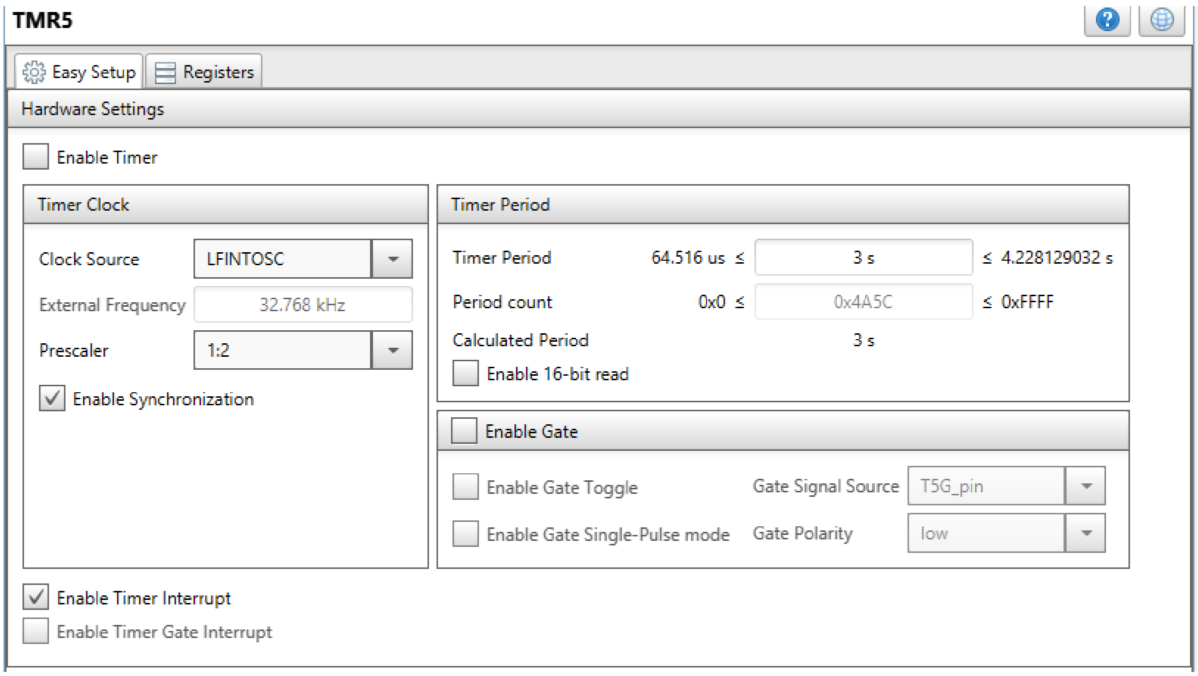
Figure 4-7. TMR2 and TMR4 MCC Configuration



TMR2 and TMR4 are configured similarly for both switch-press operations. The timer Reset source is selected through the PIN Manager tool. Pins RC5 and RB4 are configured for T2CKIPPS and T4CKIPPS, respectively, as per Curiosity HPC board switch connections.

For LED delay, TMR5 is configured in One-Shot mode to give a delay of three seconds. The timer is only initialized and not enabled. When a delay is needed, TMR5 is started in firmware. Interrupts for TMR5 are enabled to get software control for turning off the LEDs.

Figure 4-8. TMR5 MCC Configuration

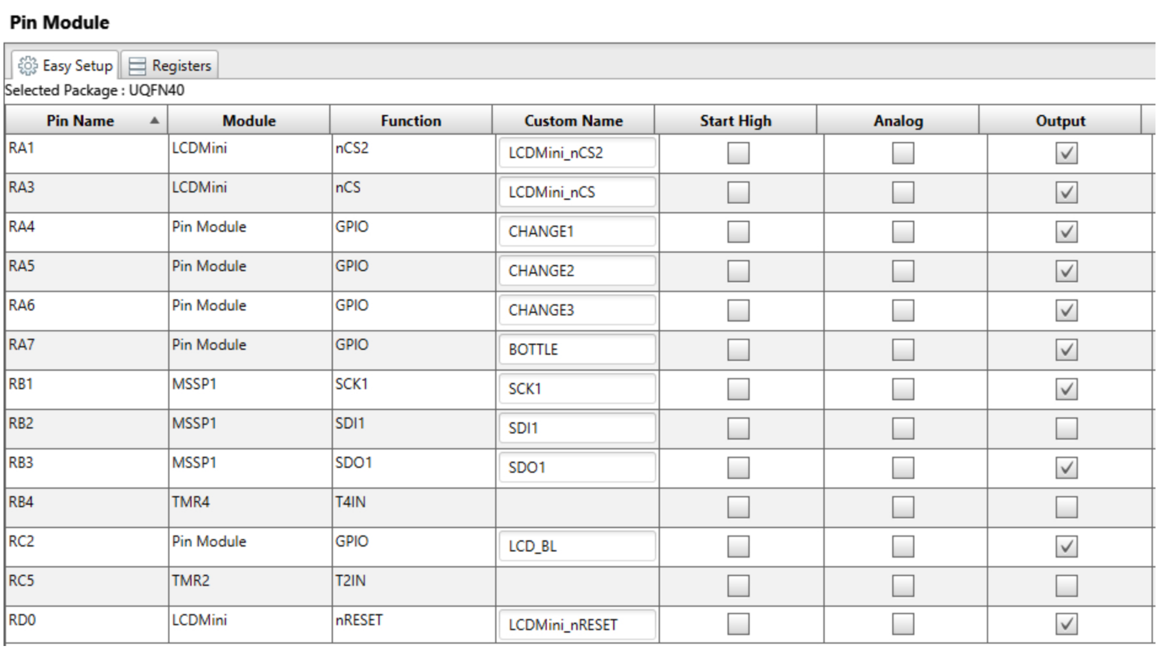


- **LCD Configuration**

  MCC supports a wide range of click boards including the **LCD mini click**. The only configuration that needs to be done is to select SPI source, Pin, and SPI speed. mikroBUS slot 1 is used to configure the LCD of the Curiosity HPC board, by selecting the corresponding pins in the Pin Manager.

  Refer to Figure 4-9 for pin configuration and Table 4-2 for pin details.

Figure 4-9. Pin Configuration for Modules



Pin Name	Module	Function	Custom Name	Start High	Analog	Output
RA1	LCDMini	nCS2	LCDMini_nCS2	☐	☐	☑
RA3	LCDMini	nCS	LCDMini_nCS	☐	☐	☑
RA4	Pin Module	GPIO	CHANGE1	☐	☐	☑
RA5	Pin Module	GPIO	CHANGE2	☐	☐	☑
RA6	Pin Module	GPIO	CHANGE3	☐	☐	☑
RA7	Pin Module	GPIO	BOTTLE	☐	☐	☑
RB1	MSSP1	SCK1	SCK1	☐	☐	☑
RB2	MSSP1	SDI1	SDI1	☐	☐	☐
RB3	MSSP1	SDO1	SDO1	☐	☐	☑
RB4	TMR4	T4IN		☐	☐	☐
RC2	Pin Module	GPIO	LCD_BL	☐	☐	☑
RC5	TMR2	T2IN		☐	☐	☐
RD0	LCDMini	nRESET	LCDMini_nRESET	☐	☐	☑

4.2.8 Firmware Overview

The vending machine demo firmware consists of minimal application code as most of the operational activities are performed by CIPs, such as CLCs and timers. The application code consists of LCD display and Sleep mode snippets.

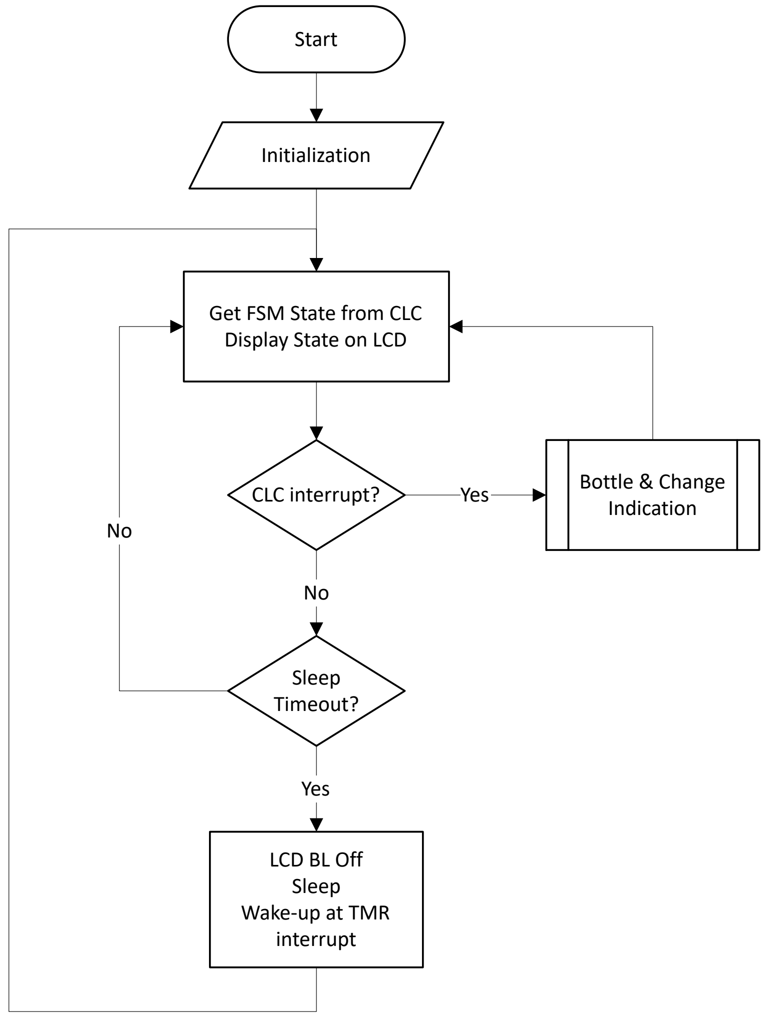
The bottle and change output are taken from CLC output and the corresponding code is written in CLC interrupt. Whenever the CLC output becomes high, the corresponding LED is turned on and TMR5 is also started, which will notify after three seconds to turn off the LEDs. This delay is also used for the LCD display interval. Therefore, a flag is used to indicate the Dispense state, which is used for LCD display function. If the system is in Dispense state, dispense messages are displayed on the LCD, otherwise the present FSM state messages are displayed. In case of an error, the CLC8 interrupt is used to set the *invalidFlag* status. This flag is used in the application firmware display routine to show the error caused by simultaneous activation of both quarter and dollar signals.

To get the CLC output for bottle and change dispensing, interrupts are enabled for the corresponding CLCs. Similarly, interrupts are enabled for TMR5 in order to turn the LED off and to clear the dispense flag timer.

This application also keeps a watch on user inactivity time. Each switch press will generate an interrupt which is used to clear the inactivity time counter and to wake up the MCU. If no switch is pressed for 60 seconds, the system will enter into Sleep mode after turning the LED back-light off. For the CLC to keep running while the CPU is in Sleep mode, the IDLE bit is set to high.

Note: The delay 60 seconds is realized using counter in while loop, thus it is approximate and will vary in case of any change in the program.

Figure 4-10. Application Flow for Vending Machine Firmware



4.2.9 Memory Requirements

Table 4-4 lists the operating conditions and peripheral configurations of the PIC18F47Q10 MCU for the vending machine application. The vending machine application firmware memory requirements are explained in Table 4-5, with MCU operating at 1 MHz of CPU clock, and for none and maximum compiler optimization levels.

Table 4-4. Peripheral Configuration for Vending Machine

MCU	PIC18F47Q10
Operating Voltage	3.3V
Operating Frequency	1 MHz
CPU Clock Configuration	1 MHz internal oscillator HFINTOSC is configured with clock divider 1
Peripherals Enabled	CLC, SPI, TMR2, TMR4, TMR5
SPI Configuration	MSSP1 is configured with 125 kHz, SPI is used for MiniLCD interface
CLC Configuration	CLC1, CLC2, CLC3, CLC4, CLC5, CLC6 CLC7 and CLC8 are configured for vending machine logic implementation
TMR2 and TMR4	250 ms delay period clocked with LFINTOSC for Monostable mode and trigger source as SW1 and SW2
TMR5	3s time delay operating in One-Shot mode

Table 4-5. Vending Machine Firmware Resource Usage

Parameters	Optimization Level	
	None (0)	Maximum (S)
Data Memory	173 bytes	91 bytes
Program Memory	4096 bytes	3196 bytes

4.3 Finite State Machine Demonstration

The vending machine implementation using the PIC microcontroller's CLC modules give a conclusive result about the simplified nature of firmware. It is elementary, unambiguous and coherent to use the CLC for logical and sequential expressions-based applications.

5. Application 2: Encoders and Decoders

In computing, data often needs to be converted from one format to another. For this purpose, encoders and decoders are needed in many applications.

5.1 Encoder

Encoding means converting the data from standard or original format to any other format; the objective of using an encoder varies from standardization, speed, secrecy, security, and/or compression of the data.

Examples:

- Rotary Encoder
- Linear Encoder
- Multiplexers
- Gray Code Encoder, etc.

5.2 Decoder

Decoding is the process of converting any encoded data back to the original data format. A decoder can take the form of a multiple-input, multiple-output, logic circuit that converts encoded input into decoded output.

Examples:

- Rotary Decoder
- Binary Decoders
- De-multiplexers
- Gray Code Decoder, etc.

5.3 Case Study – Gray Code Decoder

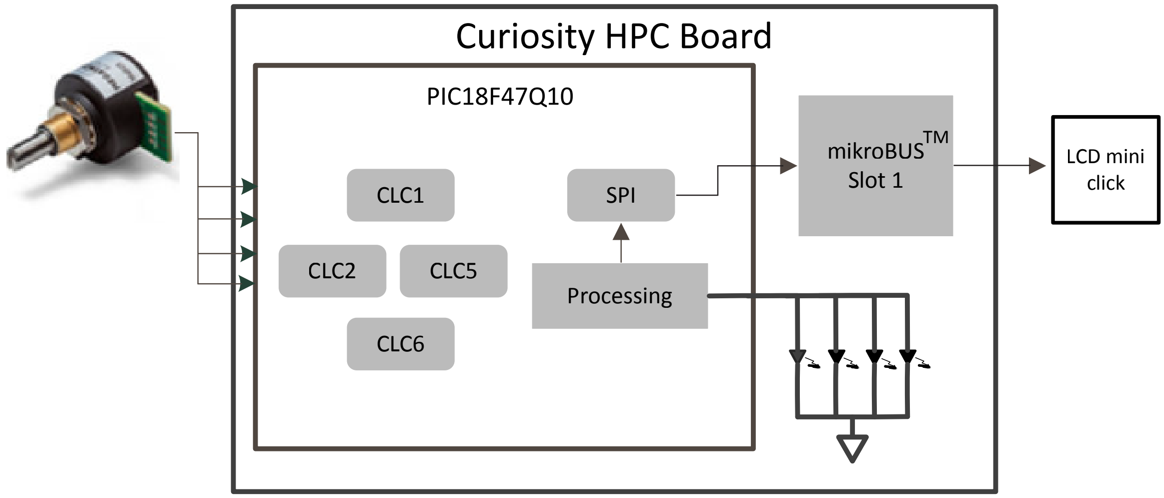
Gray code is an ordering of the binary numeral system such that two successive values differ in only one bit. More formally, a Gray code is a code assigning to each of a contiguous set of integers, or to each member of a circular list, a word of symbols such that no two code words are identical and each two adjacent code words differ by exactly one symbol. These codes are also known as single-distance codes, reflecting the Hamming distance of 1 between adjacent codes

Gray code is used in many applications such as Position encoders, Mathematical puzzle, Genetic algorithms, Boolean circuit minimization, Error correction, communication between clock domain and cycling through states with minimal effort, etc.

To use the devices operating with Gray code in digital computing systems, the decoding of data to binary is frequently required as the mathematical and logical computations cannot work. Hence, a Gray code decoder is a requirement for such applications.

The decoder can be implemented in both software and hardware. Unlike the software function, the hardware implementation is free of limitations such as input detection latency, non-real-time input detection, and time consumption.

Figure 5-1. Gray Code Decoder Using the Curiosity HPC Development Board



5.3.1 Gray Code Decoder - Logic Design Using Gates

Being a purely logical circuit, the Gray code decoder can be implemented using logic gates only. Table 5-1 describes the logical implementation of the decoder.

Table 5-1. 4-Bit Gray Code Representation from Corresponding Binary Codes

Gray Code	Binary	Decimal
0000	0000	0
0001	0001	1
0011	0010	2
0010	0011	3
0110	0100	4
0111	0101	5
0101	0110	6
0100	0111	7
1100	1000	8
1101	1001	9
1111	1010	10
1110	1011	11
1010	1100	12
1011	1101	13
1001	1110	14
1000	1111	15

Upon study of the truth table we can conclude the following points for conversion of data from Gray code to binary:

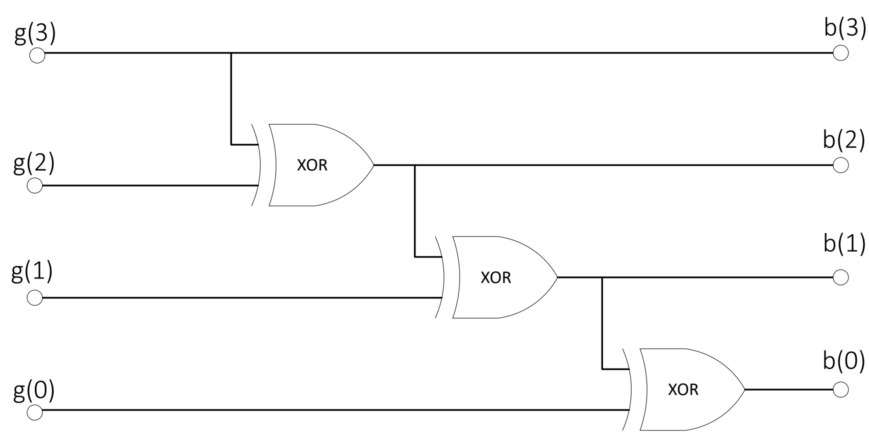
- The Most Significant bit of the Gray code number is always identical to that of the binary number.
- To derive the successive second bit, perform EX-OR operation on the Most Significant bit of the binary code and the second bit of the given Gray code.
- To derive the successive third binary bit, perform the EX-OR operation on the second Most Significant bit of the binary number to the third Most Significant bit of Gray code, and so on for the next successive binary bits conversion to find the equivalent binary code.

Also, the outputs can be solved for the Boolean expressions using K-map or other methods, to come to the similar conclusion. The derived logical expressions for Gray code decoder are as follows:

- $b(3) = g(3)$
- $b(2) = g(3) \oplus g(2)$
- $b(1) = b(2) \oplus g(1)$
- $b(0) = b(1) \oplus g(0)$

These equations can now be represented using logic gates as given in Figure 5-2.

Figure 5-2. Decoder Logic Diagram



5.3.2 Gray Code Decoder Example Using Curiosity HPC

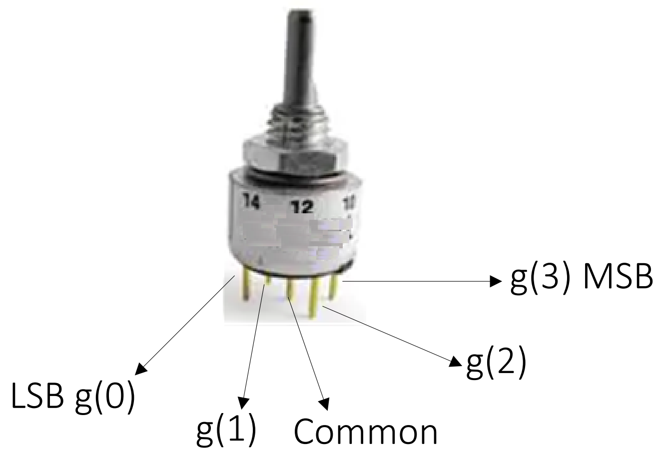
Curiosity HPC is a fully-integrated 8-bit development platform for exploring 8-bit PIC microcontrollers. Refer to 4.2.2 Vending Machine Example using Curiosity HPC for more details on the Curiosity HPC board.

5.3.3 Application Overview

In this application demo the Gray code decoder is implemented using the CLCs of the 8-bit PIC microcontroller. The 4-bit Gray code input is taken on GPIO pins and is fed to the CLCs. The CLCs will perform the logical operation and will generate the decoded output.

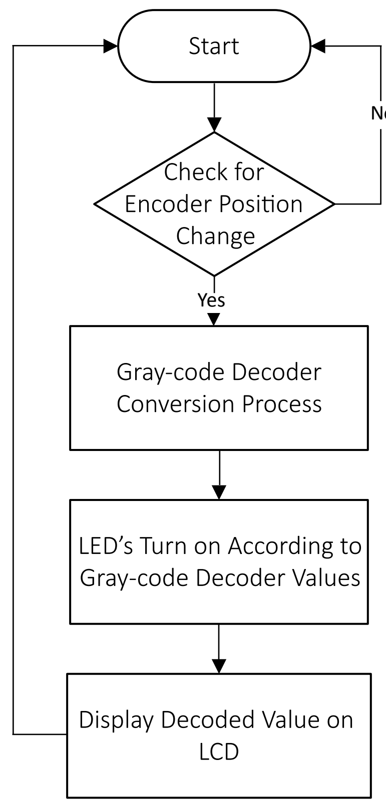
For demonstration, a 4-bit rotary encoder from Grayhill (see Figure 5-3) is used to generate Gray codes. The output from the encoder is used as input to the CLC-based decoder.

Figure 5-3. Rotary Gray Code Encoder



A logic gate-based design as shown in Figure 5-2 is implemented using CLCs configured via the MCC tool. The CLCs will perform logical operation on the encoded input values and give the decoded binary output. The output of each CLC is routed to four on-board LEDs, and is read in firmware to display on LCD. The MikroElectronika LCD mini click board™ is used to show the decoded values in the form of decimal value of the Gray code, and corresponding rotary encoder position in degrees.

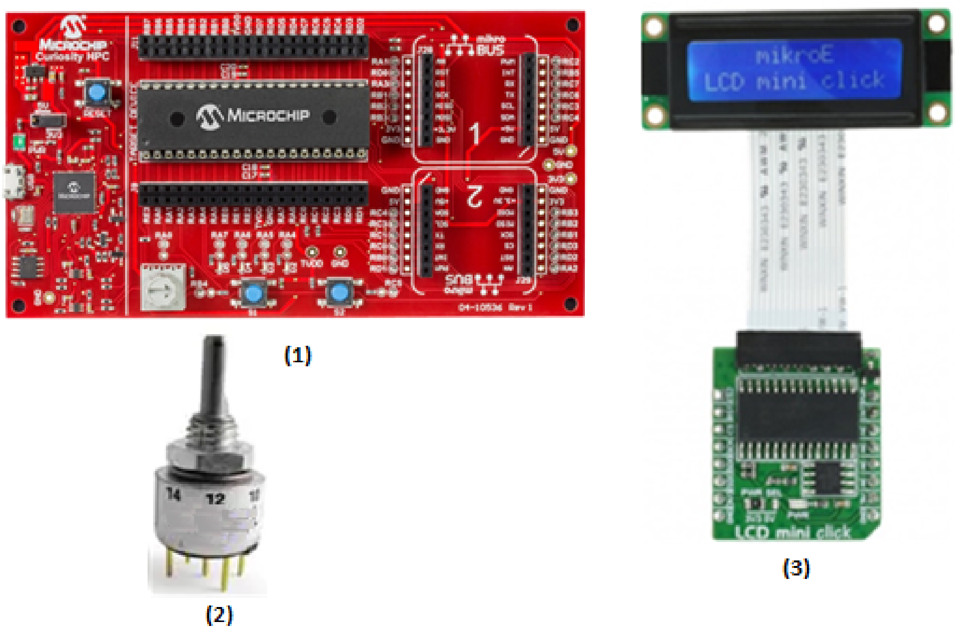
Figure 5-4. Gray Code Decoder Application Flow



5.3.4 Hardware Setup

The Curiosity HPC (1) along with the PIC18F47Q10 microcontroller is used for demonstrating the Gray code decoder application. As discussed in the application overview, the Gray code decoder application needs a Gray code rotary encoder (2) and a LCD display for displaying the decoded values by using the LCD mini click from MikroElektronika. (3).

Figure 5-5. Hardware Setup



1. Curiosity HPC Board. 2. Grayhill Rotary Encoder 3. LCD mini click

5.3.5 Pin Mapping

In the Hardware Setup section, the output of the rotary encoder is connected as input to the Gray code decoder, and the decimal output of the decoder is displayed using the LEDs and LCD. Table 5-2 shows the pin mapping of the PIC18 Q10 MCU for the Gray code decoder.

Table 5-2. Gray code Decoder Pin Mapping

Pin Function	Pin Name	Pin Type	Module Used
Decoder Input g (0)	RC0	Input-Digital	I/O
Decoder Input g (1)	RC1	Input-Digital	I/O
Decoder Input g (2)	RD2	Input-Digital	I/0
Decoder Input g (3)	RD3	Input-Digital	I/O
LCD CS	RA3	Output-Digital	LCD I/O
LCD CS2	RA1	Output-Digital	LCD I/O
LCD Reset	RD0	Output-Digital	LCD I/O
LCD BL	RC2	Output-Digital	LCD I/O
LCD SDI	RB2	Input-Digital	LCD SPI

...........continued			
Pin Function	**Pin Name**	**Pin Type**	**Module Used**
LCD SCK	RB1	Output-Digital	LCD SPI
Decoder Output b (0)	RA7	Output-Digital	I/O
Decoder Output b (1)	RA6	Output-Digital	I/O
Decoder Output b (2)	RA5	Output-Digital	I/O
Decoder Output b (3)	RA4	Output-Digital	I/O

5.3.6 Software Tools

Microchip's MPLAB X IDE, compiler and graphical code generators are used throughout the application firmware development to provide an easy and hassle-free user experience. Following are the tools used for this demo application:

- MPLAB X IDE v5.20
- XC8 Compiler v2.05
- MPLAB Code Configurator (MCC) v3.75

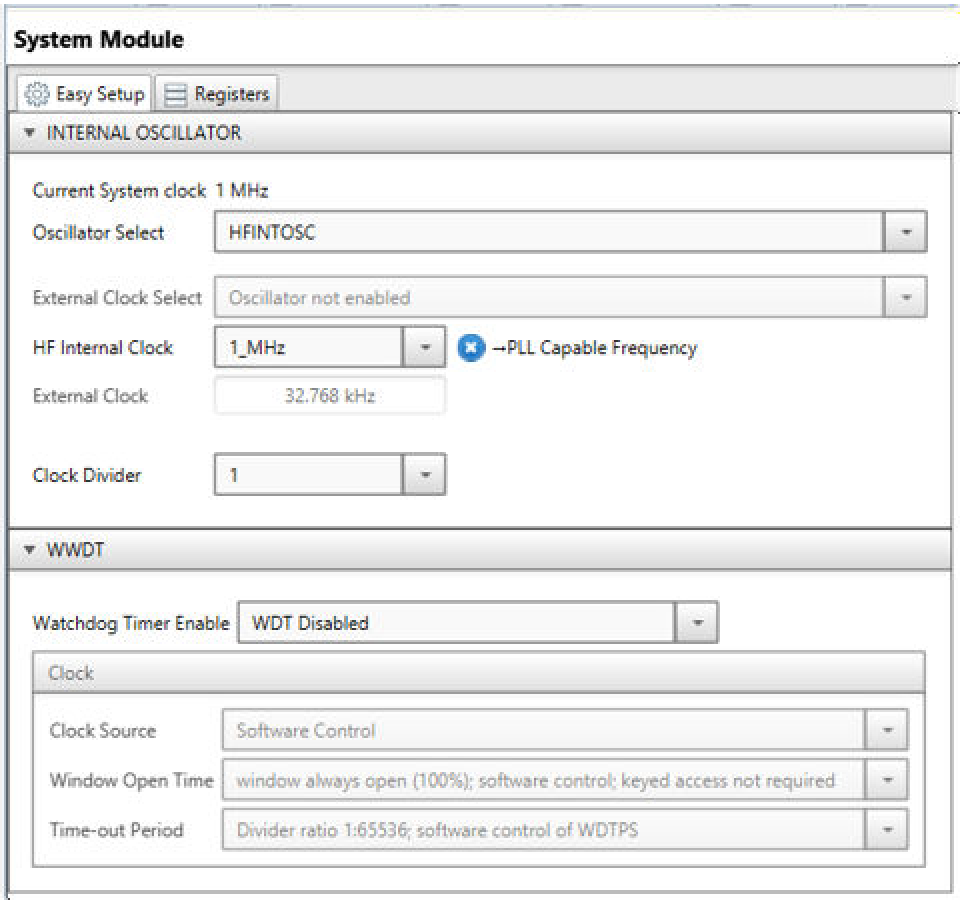
Important: For running the demo, the installed tool versions should be the same or later. This example is not tested with the previous versions.

5.3.7 MCC Configuration

- **System Configuration**

  System configuration in MCC is used for microcontroller oscillator configuration, PLL, Watchdog Timer and low-voltage programming settings. In this application, an 1 MHz oscillator with clock divider 1 is used as system clock, and Watchdog Timer disabled. Figure 5-6 shows the system configuration settings in the MCC tool.

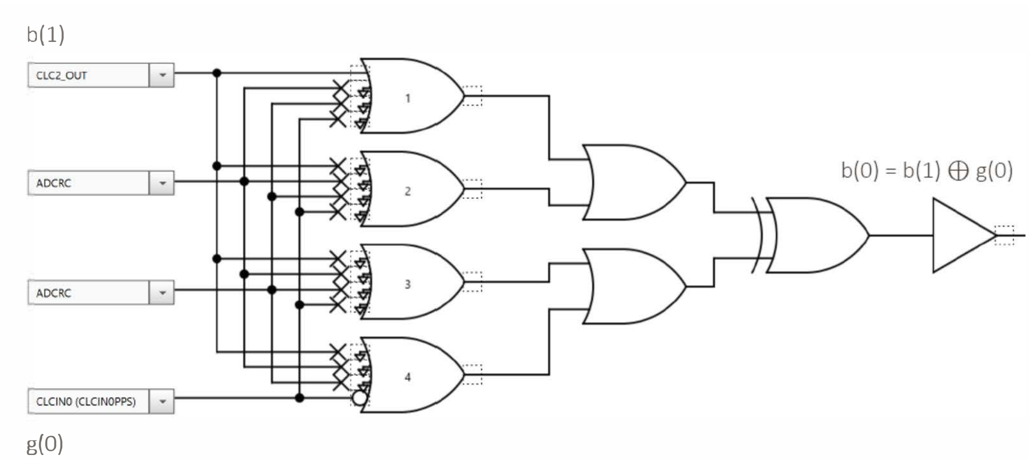
Figure 5-6. System Configuration



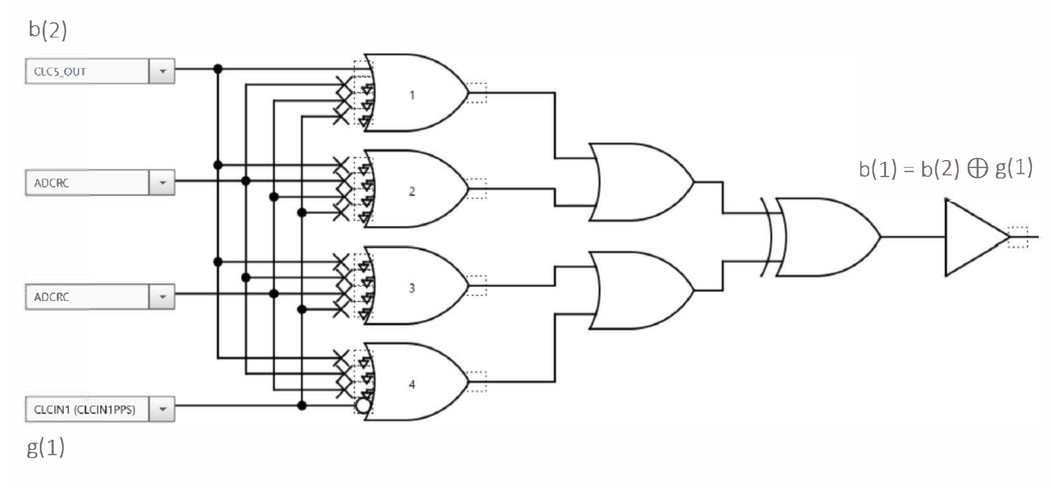
- **CLC Configuration**

  Configurable Logic Cells (CLCs) are used to decode the encoded Gray code value by performing the Boolean expressions as described in 5.3.1 Gray Code Decoder - Logic Design Using Gates. The next steps describe how the CLCs are configured for a PIC device using the MCC tool.

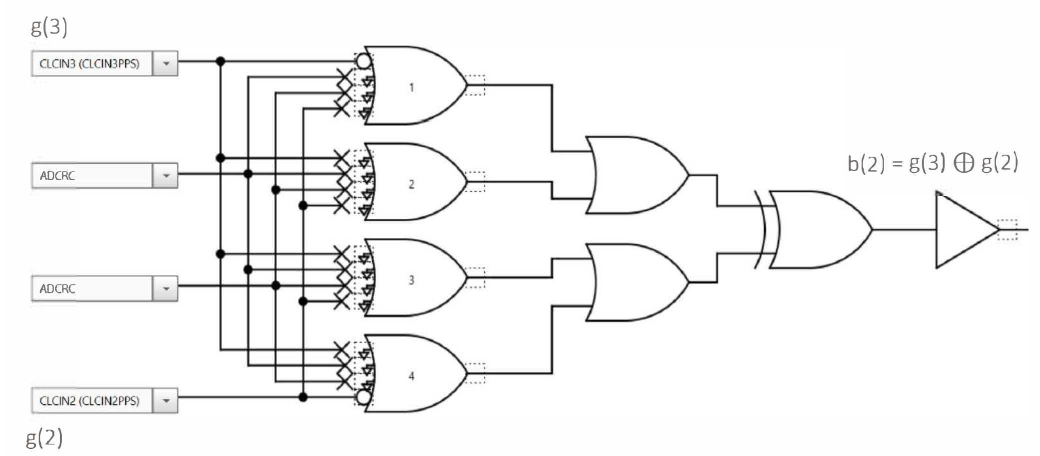
  1. CLC 1: OR-XOR



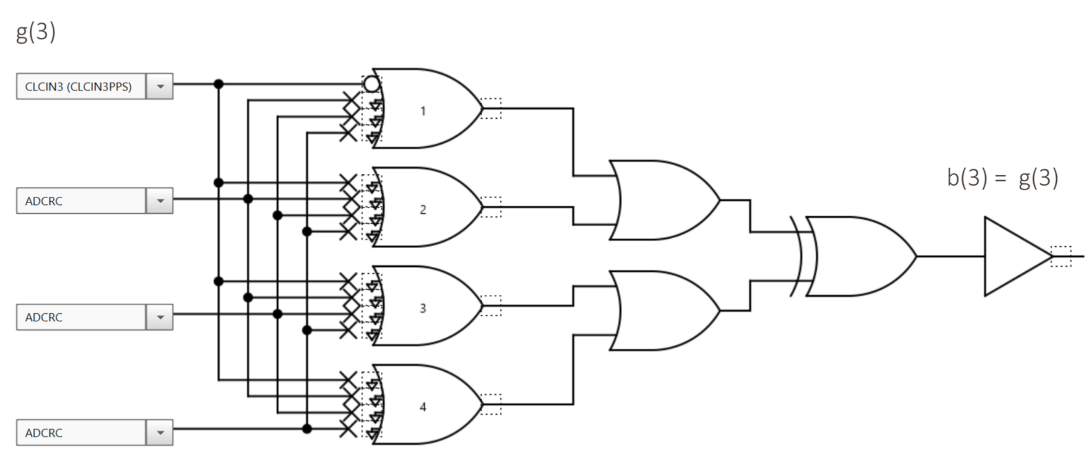
2. CLC 2: OR-XOR



3. CLC 5: OR-XOR



4. CLC 6: OR-XOR



- **LCD Configuration**

  For LCD configuration refer to the section 4.2.7 LCD Configuration.

  Pin configuration for the Gray code decoder is as per Figure 5-7.

Figure 5-7. Pin Mapping for Grey Code Decoder

Pin Module

Easy Setup | Registers

Selected Package : UQFN40

Pin Name ▲	Module	Function	Custom Na...	Start High	Analog	Output	WPU	OD	IOC
RA1	LCDMini	nCS2	LCDMini_n(	☐	☐	☑	☐	☐	none
RA3	LCDMini	nCS	LCDMini_n(	☐	☐	☑	☐	☐	none
RA4	CLC1	CLC1OUT		☐	☐	☑	☐	☐	none
RA5	CLC2	CLC2OUT		☐	☐	☑	☐	☐	none
RA6	CLC5	CLC5OUT		☐	☐	☑	☐	☐	none
RA7	CLC6	CLC6OUT		☐	☐	☑	☐	☐	none
RB1	MSSP1	SCK1	SCK1	☐	☐	☑	☐	☐	none
RB2	MSSP1	SDI1	SDI1	☐	☐	☐	☐	☐	none
RB3	MSSP1	SDO1	SDO1	☐	☐	☑	☐	☐	none
RC0	CLC1	CLCIN0		☐	☐	☐	☑	☐	none
RC0	CLC2	CLCIN0		☐	☐	☐	☑	☐	none
RC0	CLC5	CLCIN0		☐	☐	☐	☑	☐	none
RC0	CLC6	CLCIN0		☐	☐	☐	☑	☐	none
RC1	CLC1	CLCIN1		☐	☐	☐	☑	☐	none
RC1	CLC2	CLCIN1		☐	☐	☐	☑	☐	none
RC1	CLC5	CLCIN1		☐	☐	☐	☑	☐	none
RC1	CLC6	CLCIN1		☐	☐	☐	☑	☐	none
RD0	LCDMini	nRESET	LCDMini_nl	☐	☐	☑	☐	☐	
RD2	CLC1	CLCIN2		☐	☐	☐	☑	☐	
RD2	CLC2	CLCIN2		☐	☐	☐	☑	☐	
RD2	CLC5	CLCIN2		☐	☐	☐	☑	☐	
RD2	CLC6	CLCIN2		☐	☐	☐	☑	☐	
RD3	CLC5	CLCIN3		☐	☐	☐	☑	☐	
RD3	CLC2	CLCIN3		☐	☐	☐	☑	☐	
RD3	CLC6	CLCIN3		☐	☐	☐	☑	☐	
RD3	CLC1	CLCIN3		☐	☐	☐	☑	☐	

5.3.8 Firmware Overview

The Gray code decoder firmware contains minimum code. The CLCs' features accomplish the needed tasks for the application. LCD mini click and LEDs are used to display the application activities, like the binary and decimal output of the decoder and rotary encoder switch position.

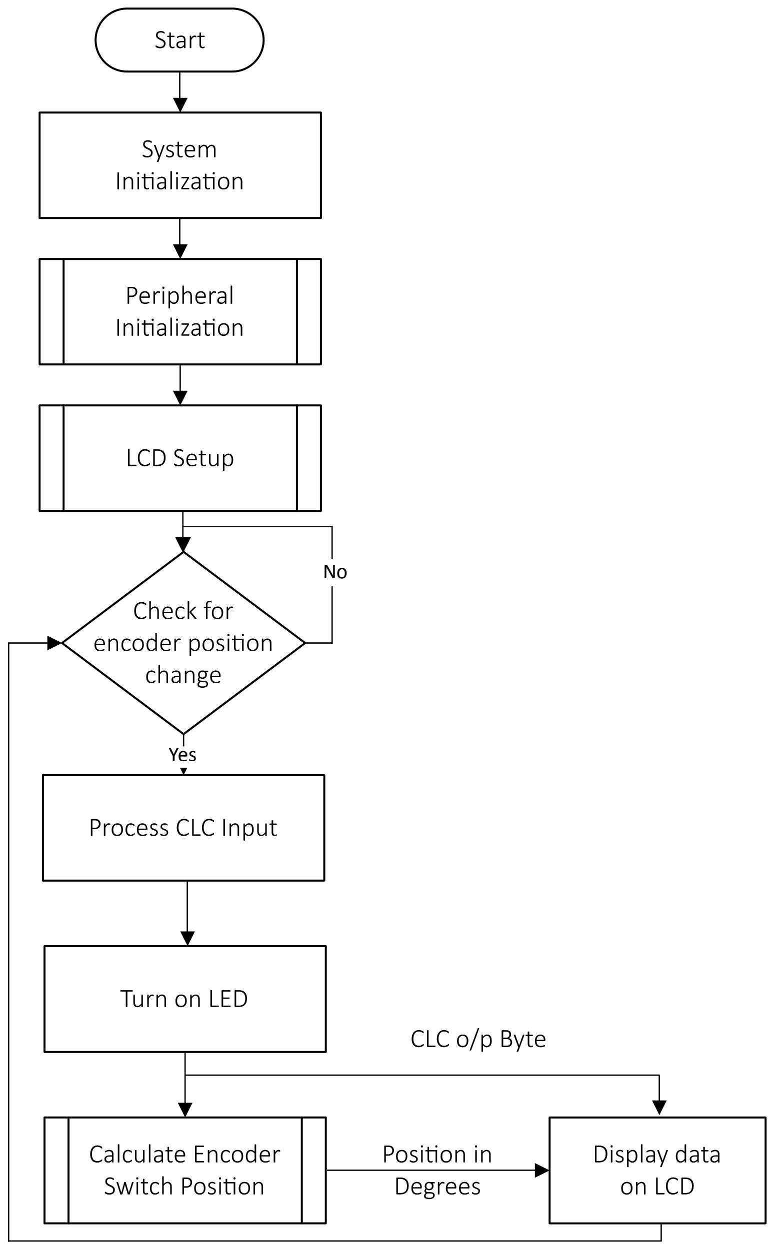
The application flow is initializing the system, pin configuration, CLC configuration and LCD display configuration. Since the CLC is a core-independent peripheral, it does not require any trigger or run command to process the input.

Input to the decoder is fed by the Gray code rotary encoder (discussed in 5.3.3 Application Overview). The decoded CLC output is available in the CLC output register (CLCDATA), as well as output lines.

These CLC output lines are routed to respective GPIOs connected to the on-board LEDs to represent the decoded values in binary format.

Since only four of the eight available CLCs on the PIC18 Q10 devices were used for designing the decoder and the CLC output bits are available in 8-bit register, shifting and masking operation is done in firmware to align the output displayed on the mini LCD in decimal format. The angular position of the encoder rotatory switch is also displayed on the LCD in degrees. The whole application flow is described in Figure 5-8.

Figure 5-8. Firmware Flow Diagram



5.3.9 Memory Requirements

The CLC requires minimum software functionality to perform Gray code decoder equations. Table 5-3 describes the operating condition and configuration of all the peripherals used. Table 5-4 and Table 5-5 show the Gray code decoder application firmware memory requirements at 1 MHz system clock.

Table 5-3. Peripherals Configured for Gray Code Decoder

MCU	PIC18F47Q10
Operating Voltage	3.3V
Operating Frequency	1 MHz
CPU Clock Configuration	1 MHz internal oscillator HFINTOSC is configured with clock divider 1
Peripherals Enabled	CLC, SPI
SPI Configuration	MSSP1 is configured with 125 kHz, and SPI is used for mini LCD click interface
CLC Configuration	CLC1, CLC2, CLC5 and CLC6 is configured without interrupt for realize the Gray code encoder equations

Table 5-4. Gray Code Decoder Firmware Specification without LCD

Parameter	Optimization Level	
	None (0)	Maximum (S)
Data Memory	72 bytes	53 bytes
Program Memory	1280 bytes	1030 bytes

Table 5-5. Gray Code Decoder Firmware Specification with LCD

Parameter	Optimization Level	
	None (0)	Maximum (S)
Data Memory	177 bytes	153 bytes
Program Memory	5318 bytes	4114 bytes

5.4 Gray Code Decoder Demonstration

The Gray code decoder implementation using the CLCs of the PIC18 Q10 family of microcontrollers shows the effective potential of CIP-based applications. As all the processing is handled by CLCs, the overall memory footprint of the code is very small; the remaining memory can be used for the balance of the user application. In conclusion, the CLCs are a very useful resource for real-time applications such as encoders and decoders.

6. Conclusion

The addition of CLCs to the Microchip set of peripherals allows users to design a logical application circuit that can interface with the PIC® microcontroller. This extends the capabilities of PIC MCU devices. Combining outputs of different peripherals and input pins using configurable gates enables and enhances the capabilities of the existing peripherals, thus expanding the horizon of applications a peripheral can accomplish.

Since the logic functions implemented in the hardware have faster event response compared to the logic functions implemented in the software, the CLC gives the advantage of faster and predictable response to users. It provides a higher level of integration without the need of external logic gates to implement the logic functions, therefore it can reduce the size of a PCB. It also helps in combining various input source signals using different logic gates to produce different signals.

There is a wide range of applications that can be implemented using CLCs; a few are discussed in this document. Microchip encourages users to explore other possibilities of using CLCs.

7. References

- Device data sheet for the specific device being used, at http://www.microchip.com.
- MCC software: http://www.microchip.com/mplab/mplab-code-configurator
- MPLAB X IDE: http://www.microchip.com/mplab/mplab-x-ide
- Demo firmware: https://mplabxpress.microchip.com/mplabcloud/example
- Curiosity HPC: https://www.microchip.com/Developmenttools/ProductDetails/DM164136
- Gray code encoder: http://www.grayhill.com/products/mechanical-encoders/
- LCD mini click: https://www.mikroe.com/lcd-mini-click

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

Trademarks

ISBN: 978-1-5224-4465-7

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

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

Atlanta
Duluth, GA
Tel: 678-957-9614
Fax: 678-957-1455

Austin, TX
Tel: 512-257-3370

Boston
Westborough, MA
Tel: 774-760-0087
Fax: 774-760-0088

Chicago
Itasca, IL
Tel: 630-285-0071
Fax: 630-285-0075

Dallas
Addison, TX
Tel: 972-818-7423
Fax: 972-818-2924

Detroit
Novi, MI
Tel: 248-848-4000

Houston, TX
Tel: 281-894-5983

Indianapolis
Noblesville, IN
Tel: 317-773-8323
Fax: 317-773-5453
Tel: 317-536-2380

Los Angeles
Mission Viejo, CA
Tel: 949-462-9523
Fax: 949-462-9608
Tel: 951-273-7800

Raleigh, NC
Tel: 919-844-7510

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110
Tel: 408-436-4270

Canada - Toronto
Tel: 905-695-1980
Fax: 905-695-2078

ASIA/PACIFIC

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

ASIA/PACIFIC

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

EUROPE

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