INTRODUCTION

A Transistor Coil Ignition (TCI) type of an Inductive Discharge Ignition (IDI) system is an electronic ignition system used in internal combustion engines. An ignition system provides a high-voltage spark in each of the engine’s cylinders to ignite the air-fuel mixture. TCI uses the charge stored in an inductor to fire the spark plug.

THE BASIC PRINCIPLE

An IDI or TCI system builds up a charge in the primary of the ignition coil then releases it at the right moment to ignite the air-fuel mixture in the cylinder. An important feature of an TCI system is the slightly longer spark duration than in a Capacitive Discharge Ignition (CDI) system.

The basic block diagram of a TCI system is shown in Figure 1.

FIGURE 1: BASIC TCI BLOCK DIAGRAM

The primary of the ignition coil is connected to a 12V battery and charged for a particular dwell time. The circuit is switched ON and OFF by a transistor/IGBT switch. When the switch is ON, the current from the primary windings in the ignition coil flow through the switch and it is grounded. The flow of current in the primary winding causes a magnetic field to be formed around the ignition coil’s secondary windings. When the switch is turned OFF, the current can no longer flow to ground and the magnetic field around the ignition coil’s secondary windings collapses. This causes a high-voltage current of 30,000 to 40,000 volts to be induced in the ignition coil’s secondary windings. This voltage is strong enough to jump the spark plug gap and generate the spark.
**DWELL TIME**

Dwell is the amount of time required to charge an inductive coil to its maximum energy level. In terms of modern engine control, dwell is defined in milliseconds. Typical ignition coils will have a dwell time between 2 ms and 5 ms. If the ignition coil is charged beyond the dwell time, it will overheat and be quickly damaged. If it is charged for a lesser duration, then it will not be able to produce a sufficient amount of spark. Dwell time puts a limitation on higher RPM, because at higher RPM the dwell time is not enough to fully charge the induction coil. That means less voltage spark at higher RPM.

**TCI USING ANGULAR TIMER**

The PIC16F161X family of 8-bit PIC microcontrollers has a core independent peripheral called Angular Timer which can be used in internal combustion engines to fire the spark at the exact firing angle with very little CPU intervention.

Figure 2 illustrates the TCI implementation using Angular Timer along with additional CIPs that are available on the PIC16F1615/9 MCU.

**FIGURE 2: TCI USING ANGULAR TIMER**

Core Independent Peripherals of PIC microcontrollers, such as Configurable Logic Cell (CLC), Angular Timer (AT), Signal Measurement Timer (SMT) and Math Accelerator are used for measuring and calculating the firing angle and dwell time control in TCI, as explained below.
**Signal conditioning circuit**

The pulser coil generates timing signals that contain both positive and negative pulses. These pulses are in the range of ±3V to ±90V, depending on the magnetic field strength of the magnet mounted on the flywheel. A signal conditioning circuit is used to invert the negative pulse and limit the pulses to a range of 0V to 5V. It is also used to filter any spurious noise. The signal conditioning circuit will provide two positive outputs, one corresponding to positive pulses and another for negative pulses. Output of the signal conditioning circuit is connected to the microcontroller.

**Configurable Logic Cell (CLC)**

The outputs of the signal conditioning circuit are connected to CLC. The outputs of the signal conditioning circuit will be in Logic 1 state, except for the positive and negative pulses from the pick-up coil. The CLC is configured as an XOR gate, so that the CLC output will give positive pulses for both the inputs, as shown in Figure 3. The CLC output is connected to the Angular Timer input internally. The Angular Timer provides a period pulse to SMT.

**FIGURE 3: SIGNAL CONDITIONING OUTPUT**

![Diagram showing signal conditioning output](image)

**Signal Measurement Timer (SMT)**

For the RPM calculation, a 24-bit Signal Measurement Timer (SMT) is used. The AT period pulse is proportional to the engine speed. Hence, it is used as input to the SMT. The SMT is configured in Windowed Measurement mode with the window input set to AT period pulse. The SMT configured in its Windowed Measurement mode can capture a value corresponding to the period of a signal. Thus, the SMT captures the value proportional to the RPM into the SMTxCPR register. Whenever the AT gives a period pulse, the SMT captures the timer value into the SMTxCPR register, resets its timer count and restarts counting. The timer capture into the SMTxCPR register generates captured period interrupt.

**ADC**

An ADC is used to find the engine temperature and the throttle position (if analog). The throttle position input can be either analog or digital. In case of a digital throttle position, the wide open throttle (WOT) will be one state and the partially open throttle (POT) will be another. A look-up table consisting of speed vs. firing angle is stored in memory. This is called a firing MAP. There will be different firing MAPs for different throttle positions and different temperature ranges.
TCI logic

RPM CALCULATION

The RPM of the engine can be calculated from the external signal frequency \( f_{\text{SIGNAL}} \) as follows:

**EQUATION 1: RPM CALCULATION**

\[
f_{\text{SIGNAL}} = \frac{\text{SMTxCLK}}{\text{SMT\_Prescalar} \times \text{SMTxCPR}}
\]

\[
\text{RPM}_{\text{Engine}} = \frac{\text{SMTxCLK}}{\text{SMT\_Prescalar} \times \text{SMTxCPR}} \times 60
\]

Where,

- \( \text{SMTxCLK} = \) clock input for SMT using SMTxCLK register = Fosc = 32 MHz
- \( \text{SMT\_Prescalar} = \) SMT clock prescaler selected in SMTx CON0 = 1:8 = 8
- \( \text{SMTxCPR} = \) the captured value of SMTxTMR at window input event
- \( f_{\text{SIGNAL}} = \) the input signal frequency to AT and SMT
- 60 scalar value is multiplied to convert Hz to RPM
- \( \text{RPM}_{\text{ENGINE}} = f_{\text{SIGNAL}} \times 60 \)

FIRING ANGLE CALCULATION

To find the firing angle for an engine between two points in the map, linear interpolation is used as follows:

**EQUATION 2: FIRING ANGLE CALCULATION**

\[
y = y_1 + (x - x_1) \left( \frac{y_2 - y_1}{x_2 - x_1} \right)
\]

Where,

- \( y = \) the instantaneous value of the firing angle to be calculated, which is in between \( y_1 \) and \( y_2 \)
- \( x = \) the instantaneous value of engine RPM, which is known and in between \( x_1 \) and \( x_2 \)
- \( (x_1, y_1) \) and \( (x_2, y_2) \) = the successive points in the graph between which a line is drawn. These are shown as row 1 and row 2 in the firing map table, respectively.
- \( x_1, x_2 = \) values of Engine RPM of two endpoints
- \( y_1, y_2 = \) values of the firing angle of two endpoints of a straight line curve

In the above equation, the ratio \( \frac{(y_2-y_1)}{(x_2-x_1)} \) is always constant and is called slope of a straight line. This can be pre-calculated for a curve to reduce the amount of calculation.

For example, Table 1 is the firing map given by engine specifications. The values in the “Slope” column are calculated from the previous two column values, and are called line slope.

FIRING MAP SELECTION

Every engine is associated with a firing map which shows the relationship between engine speeds in RPM to firing angle of the spark in degrees. These maps vary with different throttle position and engine temperature. Using the current temperature and throttle position, one of the firing maps is selected.

**FIGURE 4: EXAMPLE OF RPM TO FIRING ANGLE (THETA) MAP**

Figure 4 shows an example of a firing map. This firing map is a plot of piecewise linear curve. The red curve shows the RPM to firing angle relation for throttle ON, and the blue curve shows the RPM to firing angle relation for throttle OFF position.
EXAMPLE 1:

Suppose, RPM\(_{\text{ENGINE}} = 2400\) rpm. From Table 1, this RPM value lies between 2250 rpm with a firing angle of 16° and 2500 rpm with a corresponding firing angle of 20°. Hence, from Equation 2, the resulting terms are: \(y_1 = 16\), \(y_2 = 20\), \(x_1 = 2250\), \(x_2 = 2500\), \(x = 2400\).

\[
y = 16 + (2400 - 2250) \cdot \frac{20 - 16}{2500 - 2250} = 18.4 \approx 18°
\]

Therefore, the firing angle for the given RPM is 18°.

DWELL ANGLE CALCULATION

Dwell time is converted to phase angles corresponding to RPM and can be stored as dwell map similar to the firing map. Dwell angle value is stored into the Angular Timer Compare 1 register. The firing angle value is stored into the Angular Timer Compare 2 register.

<table>
<thead>
<tr>
<th>Engine Speed (RPM)</th>
<th>Firing Angle ((\theta))</th>
<th>Slope (m)</th>
<th>Dwell Time (mS)</th>
<th>Dwell Angle ((\theta))</th>
<th>Dwell Slope (m)</th>
</tr>
</thead>
<tbody>
<tr>
<td>250</td>
<td>4</td>
<td>0</td>
<td>5</td>
<td>8</td>
<td>0.028</td>
</tr>
<tr>
<td>500</td>
<td>4</td>
<td>0.02</td>
<td>5</td>
<td>15</td>
<td>0.04</td>
</tr>
<tr>
<td>550</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>17</td>
<td>0.02889</td>
</tr>
<tr>
<td>1000</td>
<td>5</td>
<td>0</td>
<td>5</td>
<td>30</td>
<td>0.032</td>
</tr>
<tr>
<td>1250</td>
<td>5</td>
<td>0.016</td>
<td>5</td>
<td>38</td>
<td>0.028</td>
</tr>
<tr>
<td>1500</td>
<td>9</td>
<td>0.016</td>
<td>5</td>
<td>45</td>
<td>0.032</td>
</tr>
<tr>
<td>1750</td>
<td>13</td>
<td>0.008</td>
<td>5</td>
<td>53</td>
<td>0.028</td>
</tr>
<tr>
<td>2000</td>
<td>15</td>
<td>0.004</td>
<td>5</td>
<td>60</td>
<td>0.032</td>
</tr>
<tr>
<td>2250</td>
<td>16</td>
<td>0.016</td>
<td>5</td>
<td>68</td>
<td>0.028</td>
</tr>
<tr>
<td>2500</td>
<td>20</td>
<td>0</td>
<td>5</td>
<td>75</td>
<td>0.03067</td>
</tr>
<tr>
<td>3250</td>
<td>20</td>
<td>0.008</td>
<td>5</td>
<td>98</td>
<td>0.028</td>
</tr>
<tr>
<td>3500</td>
<td>22</td>
<td>0</td>
<td>5</td>
<td>105</td>
<td>0.03</td>
</tr>
<tr>
<td>7500</td>
<td>22</td>
<td>0</td>
<td>5</td>
<td>225</td>
<td>0</td>
</tr>
</tbody>
</table>
EXAMPLE 2: DWELL AND FIRING ANGLE CALCULATION

Dwell time can be converted to dwell angle using the following equation:

\[ D = \frac{\text{dwell time in ms} \times 0.001 \times 360^\circ}{\text{period of pick-up signal}} = \frac{\text{dwell time in ms} \times 0.001 \times 360^\circ}{(\text{RPM})/60} \]

Suppose, RPM\text{ENGINE} = 2000 \text{ rpm}
The angle between advance pulse and TDC = 39°
From Table 1, the RPM value has a firing angle of 20°
The dwell angle for 5 mS dwell time and 2000 RPM is

\[ D = \frac{5 \times 0.001 \times 360^\circ}{2000/60} = 60^\circ \]

So, the IGBT should be turned ON for 60° before the firing event.

If an angle of 0° is considered at advance pulse, then the value to insert in the Compare 1 register for starting the dwell time is:

\[ ATCCY1 = 39 - 20 - 60 = -41^\circ = 360 - 41^\circ = 319^\circ \]

And for the firing event the value to insert in the Compare 2 register is:

\[ ATCCY2 = 39^\circ - 20^\circ = 19^\circ \]

MATH ACCELERATOR

The Math Accelerator peripheral, also called the PID module, is used to calculate the firing angle and the dwell angle for current RPM using linear interpolation, as shown in Equation 2. Using the Math Accelerator the calculation can be done faster. It performs an addition and a multiplication as shown below:

EQUATION 3:

\[ R = (A + B) \times C \]

Where,
A = the first input operand for addition (PIDxIN register)
B = the second input operand for addition (PIDxSET register)
C = the third input operand for multiplying with result of addition (PIDxK1 register)
R = the results of one complete add and multiply operation (PIDxOUT register)

The Math Accelerator performs these operations in 16-bit integer, either in signed or unsigned format configured by the user. The result of this operation is 32-bit in size, stored into the PIDxOUT register. The result of this operation can be configured to be accumulated or non-accumulated. For accumulation of the result, the PIDxACC register is used to accumulate previous results.

ANGULAR TIMER

The dwell and the firing angle values are calculated using the Math Accelerator. Dwell angle value is stored into the Angular Timer Compare 1 register. The firing angle value is stored into the Angular Timer Compare 2 register.
When the value in the Compare 1 register matches the AT phase counter, a compare interrupt is generated and the IGBT is turned ON, as shown in Figure 5. This initiates the charging of the primary of an ignition coil. The primary of the ignition coil is charged for a particular dwell time. When the value in the Compare 2 register matches the AT phase counter, a compare interrupt is generated and the IGBT is turned OFF to disconnect the primary of the ignition coil. At this firing event the magnetic field around the ignition coil’s secondary windings collapses and the spark is generated.

**DEBUG MODE**

In case of a Debug mode, a connection to the PC GUI can be provided using UART. The firing MAP or specific parameters like PIP distance or dwell time can be changed on the fly using a PC-based GUI and an USB to UART converter. For details regarding the engine tuning GUI, refer to Appendix B: “Debug Mode and Engine Tuning PC GUI”.

---

**FIGURE 5: TCI TIMING DIAGRAM**

![TCI Timing Diagram](image)
TCI INTEGRATED WITH RKE, BIKE FINDER AND IMMOBILIZER

Using Core Independent Peripherals (CIPs) for the implementation of a TCI system frees up the CPU time and also reduces the code size which can be utilized for adding different enhancement features to the system, such as Remote Keyless Entry (RKE) (i.e., anti-theft remote lock, bike finder and immobilizer, etc.). The following section explains the implementation of a TCI system integrated with RKE, bike finder and immobilizer functionalities.

FIGURE 6: TCI INTEGRATED WITH RKE, BIKE FINDER AND IMMOBILIZER
Remote Keyless Entry (RKE) and Bike Finder System

The Remote Keyless Entry (RKE) system is an electronic lock that controls the access to a vehicle without using a traditional mechanical key. Widely used in automobiles, an RKE performs the functions of a standard mechanical key without physical contact. When within a few yards of the vehicle, pressing a button on the remote keyfob can lock or unlock the vehicle, and may perform other functions like vehicle finder. When the remote lock is switched on, the vehicle could be started only when the remote lock is opened or unlocked and the ignition keys are turned ON.

The RKE and the bike finder system consists of two blocks: a keyfob or remote, and an RF receiver on the vehicle side.

FIGURE 7: 64-BIT KeeLoq ENCRYPTED DATA

```
// keeloaq | receive buffer map
//
// 8 7 6 5 4 3 2 1 0
//
// I-S/N -> SERIAL NUMBER  (20 BIT)
// K=KEY -> buttons encoding (4 BIT)
// S=Sync -> Sync counter    (16 BIT)
// D=Disc -> Discrimination bits (10 BIT)
// R=Repeat -> Repeat/first (1 BIT)
// V=Vlow -> Low battery    (1 BIT)
```

RF RECEIVER IN VEHICLE SIDE

The vehicle side consists of an RF receiver, where the data sent by the keyfob is captured. The microcontroller then decodes the captured data. It then sends an appropriate message to lock or unlock the vehicle. If the vehicle is parked in a crowded parking lot, the bike finder key can be pressed to locate the vehicle. If the bike finder key is pressed, then flash lights on the vehicle can be lit and blinked along with the buzzer beep, so that the user can easily locate the vehicle.

RF receiver

An RF receiver, MICRF229, is used in this design. For the details on how to implement the RF receiver section, refer to “MICRF229-433 Evaluation Board User Guide”.

The RF receiver to microcontroller interface is shown in Figure 8.
Immobilizer or Passive Keyless Entry (PKE)

An immobilizer is an electronic security device fitted to an automobile to prevent the engine from running, unless the correct key or other token is present. This prevents the vehicle’s theft.

Whenever a keyfob comes in proximity of a vehicle, the microcircuit inside the key is activated by a small electromagnetic field produced by a coil in the immobilizer unit on the vehicle. This induces current to flow inside the key body, and the keyfob is powered. The keyfob then broadcasts a unique binary code. This code is read by the automobile’s immobilizer unit. When the immobilizer unit determines that the coded key is both current and valid, it enables the ignition.

The immobilizer or Passive Keyless Entry (PKE) can be implemented using two different methods:

1. Using 125 kHz Low Frequency (LF)
2. Using 13.56 MHz High Frequency (HF)

IMMOBILIZER WITH 125 kHz (LF)

The immobilizer unit at the vehicle’s side produces a magnetic field of 125 kHz periodically.

Keyfob uses a HCS410 device, which is a low-frequency (LF) transponder (battery less) plus a KEELQ® encoder in a single chip. The coil in the immobilizer unit is part of the Resistance Inductor Capacitor (RLC) circuit. The immobilizer unit communicates to the transponder by switching the 125 kHz signal to the series RLC circuit ON and OFF. Thus, the immobilizer unit’s magnetic field is switched ON and OFF. The transponder coil is connected in parallel with a resonating capacitor (125 kHz) and a KEELQ HCS410 transponder integrated circuit. When the transponder in the keyfob is brought into the proximity of the immobilizer unit’s magnetic field, it magnetically couples with it and draws energy from it. Even if the battery in the keyfob goes flat, the HCS410 will still be able to get power from the field generated by the immobilizer unit’s coil. The voltage on the coil can reach over 400 VPP and has a peak current of 1A.

The keyfob then broadcasts a unique binary key code. This code is read by the automobile’s immobilizer unit. When the immobilizer unit validates the key, it enables the ignition and the vehicle is mobilized.

IMMOBILIZER WITH 13.56 MHz HIGH FREQUENCY (HF)

The immobilizer unit at the vehicle side produces a magnetic field of 13.56 MHz periodically. The immobilizer unit at vehicle side has a 13.560 MHz high-frequency loose transformer circuit, as shown in Figure 9. The transformer can be formed using a PCB antenna, also called a coil. Its function is to supply energy to the keyfob, in case there is valid low-power keyfob condition detection. The 13.560 MHz oscillator and amplifier have to condition the signal for properly driving the HF supply transformer. A low voltage condition can be sent as one of the commands in the keyfob transmitter side, when that condition exits, alerting the user to replace the battery. To obtain maximum performance, a band-pass filter and a matching network are used.
The keyfob uses a PIC16F microcontroller along with the MICRF112 RF 433 MHz transmitter, as shown in Figure 10. The 13.56 MHz high-frequency (HF) transformer uses loose coupling transformer techniques to harvest energy from another similar type of transformer (immobilizer unit side) to supply energy to the keyfob in a low battery (less than 1.8 volts) condition. This part of the circuit ensures operation in a condition with low voltage battery. The matching network and low pass filter circuits are needed to obtain maximum power efficiency and get rid of undesired harmonics and spurs. The LDO ensures that no high voltage from the high-frequency transformer goes to the rest of the circuit.
The advantages of using a 13.56 MHz frequency instead of 125 kHz for energy harvesting are the following:

1. In case of a 125 kHz system, the voltage on the coil at immobilizer side can reach over 400 VPP and has a peak current of 1A for generating the strong magnetic field. Whereas a 13.56 MHz system is a low-power one.

2. Air core transformer winding is required in case of a 125 kHz system. Whereas, for the HF transformer the PCB antenna will be enough. The size of the PCB antenna required for a HF transformer implementation with 13.56 MHz will be significantly reduced compared to the 125 kHz system.

3. The overall cost of the immobilizer unit, which uses a 13.56 MHz energy harvesting scheme, will be smaller than with a 125 kHz system.
FIRMWARE

The firmware flowchart is shown in Figure 11.

**FIGURE 11: FIRMWARE FLOWCHART**

Program Flow

The program execution is divided into two major parts:

1. **The RF receiver section with KEELOQ decoding.**
2. **The TCI ignition system section.**

The Reset state will be Locked state. It will wait for the unlock key press to start the ignition. If a bike finder key is pressed, then the signals to blink the flash lights on the vehicle and to beep the buzzer are sent. When the unlock key press is authenticated, the ignition system will start functioning.

For KEELOQ decoding algorithms refer to application note AN744, “Modular Mid-range PICmicro® KEELOQ® Decoder in C” (DS00744).

The ignition part of the firmware is sequentially divided into steps to compute the RPM and firing angle as explained below:

1. **Initialize:** The board is powered on or reset; all the peripherals (i.e., ADC, PID, CWG, CLC1, SMT1 and AT1 are initialized). Please refer to Appendix C for initialization of peripherals using MPLAB® Code Configurator (MCC).
2. **Input Acquisition:** The CLC1 takes two inputs to generate one XOR-ed output connected to the AT input. AT starts counting and gives period pulses. This AT period pulse is used by the SMT as window input in the Windowed Measure mode. The SMT at every AT period input captures the value of the SMT1TMR register to SMT1CPR, and a corresponding interrupt is generated. The program is notified about the new SMT value available for calculation. This is the acquisition step.
3. **ADC:** The engine temperature and throttle position sensor’s analog output is digitized by the internal 10-bit ADC.
4. **RPM calculation**: The engine RPM is calculated from the SMT1CPR register value.

5. **MAP selection**: Using temperature and throttle position values, a particular MAP is selected. From this selected MAP, the firing angle corresponding to the present RPM is read from the look-up table. If the current RPM is in between the values stored in the look-up table, then the firing angle and the dwell angle for that RPM is calculated using interpolation.

6. **Dwell and firing angle update**: The dwell angle and the firing angle values are stored in the Angular Timer Compare 1 registers (AT1CC1) and Compare 2 registers (AT1CC2), respectively.

7. **Angular Timer Compare interrupt 1**: Upon Angular Timer Compare 1 event, Logic 1 is given to the port pin connected to the IGBT gate. The IGBT is turned ON to start the charging of the ignition coil.

8. **Angular Timer Compare interrupt 2**: Upon Angular Timer Compare 2 event, Logic 0 is given to the port pin connected to the IGBT gate. The IGBT is turned OFF, so that the charging of the ignition coil is interrupted and the spark is generated.

9. Steps 2 to 8 are repeated at every AT period cycle.

**COMPARISON BETWEEN TCI USING ANGULAR TIMER AND TCI USING THE TRADITIONAL APPROACH**

**Firing Angle to Timer Counts Conversion**

For the traditional method, the firing angle has to be converted to corresponding timer counts.

Whereas, the firing angle value obtained from the MAP can be directly written to the Angular Timer Compare register. When the angle value matches, it generates an interrupt and the IGBT can be turned OFF. There is no CPU intervention, because the Angular Timer is a separate on-chip peripheral.

For the TCI system, if using the Angular Timer, it is needed to convert the dwell time to corresponding angle for every RPM.

**Interpolation Calculation for Firing Angle**

For the traditional method, the interpolation calculation of the firing angle and the conversion of the firing angle to timer count calculation take longer.

By using the Math Accelerator, the firing and the dwell angle interpolation calculations can be expedited.

**RPM calculation**

When applying the traditional method, for lower RPMs there is the possibility of timer overflow.

SMT is a 24-bit timer. If used for RPM calculations with clock frequency of 1 MHz, it can count till 0.125 Hz (i.e., 7.5 RPM without overflow).

**Resolution**

With the traditional method, the angular resolution varies depending on the RPM.

Whereas, the desired resolution of the firing angle can be directly loaded into the resolution register (ATxRES) of the Angular Timer. For a resolution of 1° the resolution register is loaded with a value of 359 in case of Single Pulse mode. The angular resolution is independent of the engine RPM.

**Overall Comparison**

The maximum frequency of the input pick-up signal to the TCI system can be 190 Hz corresponding to 11,400 RPM (i.e., 190 Hz (period 5.263 ms)). Within this period of 5 ms, the TCI operation must finish all the execution and calculations. Considering the PIC16F devices for both traditional method and using the Angular Timer, which has 8 MIPS of execution speed at 32 MHz, the comparison table below shows the CPU usage for both systems.

The MIPS can be calculated as in Equation 4:

\[
\text{MIPS}_{\text{Actual}} = \frac{\text{MIPS}_{\text{Total}} \times T_{\text{Execution}}}{T_{\text{MIN}}} 
\]

Where,

- MIPS\_Actual = the calculated MIPS for current program execution
- MIPS\_Total = the max MIPS of the device
- TMIN = the minimum input period which is 1/fMAX, i.e. inverse of minimum input signal frequency
- TEXECUTION = the actual execution time of program

In this case, the TMIN = 5 ms and the MIPS\_TOTAL = 8.

Thus, CPU usage based on max MIPS is given in Equation 5:

\[
\%\text{CPU Usage} = \frac{\text{MIPS}_{\text{Actual}} \times 100}{\text{MIPS}_{\text{Total}}} 
\]
Table 3 summarizes the resource comparisons of two methods for TCI implementation and TCI with integrated RKE, bike finder and immobilizer:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>TCI using Angular Timer</th>
<th>TCI using Traditional Method</th>
<th>TCI with Integrated RKE, Bike Finder and Immobilizer</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flash Memory (Words)</td>
<td>2226</td>
<td>3502</td>
<td>3265</td>
</tr>
<tr>
<td>RAM (Bytes)</td>
<td>80</td>
<td>270</td>
<td>345</td>
</tr>
<tr>
<td>Peripherals Used</td>
<td>ADC, CLC, AT, SMT, Math</td>
<td>ADC, CCP, Timer1</td>
<td>ADC, CLC, AT, SMT, Math Accelerator, Timer</td>
</tr>
</tbody>
</table>

Table 4 summarizes the performance comparisons of two methods for CDI implementation:

<table>
<thead>
<tr>
<th>Parameters</th>
<th>TCI using Angular Timer</th>
<th>TCI Using Traditional Method</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>Depends on FOM and ATxRES value</td>
<td>Depends on engine speed (i.e., RPM and granularity of timer count for firing angle calculation).</td>
</tr>
<tr>
<td>Resolution</td>
<td>Minimum 0.35° resolution (i.e., 10-bit ATxRES value)</td>
<td>Depends on Timer1 clock and RPM. E.g., for 1 MHz clock and 250 RPM the angular resolution is 0.09°, and for 10,000 RPM the angular resolution is 3.6°.</td>
</tr>
</tbody>
</table>
CONCLUSION

The TCI system can be implemented using PIC16F microcontrollers either by applying the traditional method or by using CIPs like AT, CLC, SMT, Math Accelerator and CWG. However, using the CIPs greatly improves the overall performance and implementation of the TCI. The AT successfully divides the input signal to angular division without CPU intervention, which helps to boost the performance and hence, removing the need for firing angle conversion from degree to equivalent time. These angular divisions are also accurate and constant throughout the input signal range. As shown in the comparison Table 2, Table 3 and Table 4, the performance of the TCI system can be greatly enhanced using CIPs like Angular Timer, Math Accelerator and SMT, etc. In addition, with the use of the Math Accelerator, the calculations are now more accurate and faster. The SMT with its inherent high-bit resolution helps in tracking low engine RPMs and taking necessary action, without the need for large computations. Hence, the remaining CPU bandwidth can be used for adding other enhancement features like anti-theft remote lock (RKE), bike finder and immobilizer.
APPENDIX A: SCHEMATICS/PCB FILES AND REFERENCE DESIGN

For schematics/PCB files and reference board, contact the local Microchip sales office.

FIGURE A-1: TCI BOARD

RF receiver
APPENDIX B: DEBUG MODE AND ENGINE TUNING PC GUI

This appendix explains the PC-based GUI application, which is developed for debugging and engine tuning of a TCI/CDI system.

As explained in the previous sections, microcontroller-based TCI/CDI applications have a timing/firing map/look-up table to handle the advance of the ignition depending upon the engine speed, temperature and throttle position. These are based on the ignition timing advance characteristics specified by the engine manufacturers. Tuning a TCI/CDI system by varying firing angles at different RPMs aids in checking and improving the efficiency, thereby enhancing the performance of the engine.

Installation and Usage of the Desktop Application

1. Start the IgnitionSystemTuning.exe application. Figure B-1 shows the GUI of the tuning application.
As shown in Figure B-1, in the top left corner of the GUI is the Engine Settings box. The Settings dialog box requires the user to enter the engine parameters and the Angular Timer parameters for MAP generation and tuning. These values can be changed by the user at any time.
2. On the left side, the GUI has a table consisting of three columns:
   • Engine speed in RPM
   • Firing Angle in Degrees
   • Dwell Time in µs

   In the middle, there are two text boxes labeled “Application Messages” and “Application Debug Information”. “Application Messages” displays errors and any message during operation. “Application Debug Information” displays the real-time engine parameters like engine speed, firing angle, dwell angle, temperature and throttle position etc.

3. The application has a menu labeled “Application” on the topmost left corner. Upon clicking on it, it displays various options, such as “Clear All” and “File”. “Clear All” menu option clears all the text boxes and the user details of the engine in Settings and the table. It resets the GUI to its default state. The “File” menu has two options: Save and Load. The Save option saves the engine information entered by the user in a file at a user-specified location. The Load option loads the engine information file from a user-defined location.

4. The Application has a menu labeled “Debug”. To start the debug messages click on the Debug menu and select “ON”. The application sends a request to the TCI/CDI board to send the real-time ignition data to the user for display. To stop the debug messages getting displayed on the GUI, click on the Debug menu and select “OFF”. The application requests the TCI/CDI board to stop the transmission of real-time debug messages to the GUI.

5. The functions of various buttons on the GUI are listed below:
   LOAD MAP – When this button is clicked, the application finds any connected TCI/CDI board and starts sending the tuning MAP data and the engine parameters set by the user. If the data transmission is successful, a success message is displayed. If the data transmission is not successful, an error message will appear. If an error occurs, it tries to resend the data two more times.

   START – When this button is clicked, the tuning operation with the TCI/CDI board will begin. The board sends real-time data and it is displayed on the GUI.

   STOP – When this button is clicked, the tuning operation with the TCI/CDI board is stopped and the board is reset for normal operation.

6. The right side of the GUI shows the animated real-time data. During the tuning process, data such as IGBT angle turn ON, IGBT angle turn OFF, temperature and throttle position, is displayed visually, as well as in text format, as shown in Figure B-3.
Figure B-3 shows the GUI screen during the tuning process.
APPENDIX C: MPLAB® CODE CONFIGURATOR (MCC)

The MPLAB® Code Configurator (MCC) plug-in for MPLAB X can be used to configure the peripherals in the PIC® MCU. The MCC provides graphical user interface (GUI) tools to easily understand the configuration and select the required settings. This reduces the time to development. The MCC also provides some built-in functions for working with specific peripherals.

To install MCC, go to Tools → Plugins Available Plugins → MPLAB X Code Configurator → Install. To start and use the MCC after installation, go to Tools → Embedded → MPLAB Code Configurator.

Figure C-1 to Figure C-6 shows the configuration of the PIC16F1619 device peripherals such as AT, CLC, Math Accelerator, SMT and EUSART, respectively, used in the TCI implementation.

FIGURE C-1: AT CONFIGURATION IN MCC FOR PIC16F1619
FIGURE C-2: CLC CONFIGURATION IN MCC FOR PIC16F1619

FIGURE C-3: MATH ACCELERATOR CONFIGURATION IN MCC FOR PIC16F1619
FIGURE C-4: SMT1 CONFIGURATION IN MCC FOR PIC16F1619

FIGURE C-5: EUSART CONFIGURATION IN MCC FOR PIC16F1619
After all the settings are selected in MCC, click the Generate Code button to generate the peripheral code, and create a header and a C file for every peripheral separately. Figure C-6 shows the files created and arranged under the MCC Generated Files folder for TCI implementation using AT. The programmer should add the application code wherever it is required in these files.
APPENDIX D: REFERENCES

1. PIC16(L)F1615/9 14/20-Pin, 8-Bit Flash Microcontroller Data Sheet (DS40001770)
2. RF Receiver MICRF229 Data Sheet
3. MICRF229-433 Evaluation Board User’s Guide
4. AN744, Modular Mid-Range PICmicro® KEELOQ® Decoder in C (DS00744)
5. TB003, An Introduction to KEELOQ® Code Hopping (DS91002)
6. TB001, Secure Learning RKE Systems Using KEELOQ® Encoders (DS91000)
7. HCS410 KEELOQ® Code Hopping Encoder and Transponder Data Sheet (DS40158)
8. AN650, Designing a Transponder Coil for the HCS410 (DS00650)
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.

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.

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.

QUALITY MANAGEMENT SYSTEM CERTIFIED BY DNV

ISO/TS 16949

TRADEMARKS

The Microchip name and logo, the Microchip logo, AnyRate, dsPIC, FlashFlex, flexPWR, Hedo, JukeBlox, KEELOQ, KEELOQ logo, Kleer, LANCheck, LINK MD, MediaLB, MOST, MOST logo, MPLAB, OptoLyzer, PIC, PICSTART, PIC32 logo, RightTouch, SpyNIC, SST, SST Logo, SuperFlash and UNI/O 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.


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 trademarks 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.

© 2016, Microchip Technology Incorporated, Printed in the U.S.A., All Rights Reserved.
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

Cleveland
Independence, OH
Tel: 216-447-0464
Fax: 216-447-0643

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

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

New York, NY
Tel: 631-435-6000

San Jose, CA
Tel: 408-735-9110

Canada - Toronto
Tel: 905-673-0699
Fax: 905-673-6509

ASIA/PACIFIC
Asia Pacific Office
Suites 3707-14, 37th Floor
Tower 6, The Gateway
Harbour City, Kowloon

Hong Kong
Tel: 852-2943-5100
Fax: 852-2941-3431

Australia - Sydney
Tel: 61-2-9868-6733
Fax: 61-2-9868-6755

China - Beijing
Tel: 86-10-8569-7000
Fax: 86-10-8528-2104

China - Chengdu
Tel: 86-28-8665-5511
Fax: 86-28-8665-7889

China - Chongqing
Tel: 86-23-8980-9500
Fax: 86-23-8980-9560

China - Dongguan
Tel: 86-769-8702-9880
Fax: 86-769-8702-9880

China - Hangzhou
Tel: 86-571-8792-8115
Fax: 86-571-8792-8116

China - Hong Kong SAR
Tel: 852-2943-5100
Fax: 852-2941-3431

China - Nanjing
Tel: 86-25-8473-2460
Fax: 86-25-8473-2470

China - Qingdao
Tel: 86-532-8502-7355
Fax: 86-532-8502-7205

China - Shanghai
Tel: 86-21-5407-5533
Fax: 86-21-5407-5066

China - Shenyang
Tel: 86-24-2334-2829
Fax: 86-24-2334-2393

China - Shenzhen
Tel: 86-755-8864-2200
Fax: 86-755-8203-1760

China - Wuhan
Tel: 86-7-5980-5300
Fax: 86-7-5980-5118

China - Xian
Tel: 86-29-8833-7252
Fax: 86-29-8833-7256

ASIA/PACIFIC
China - Xiamen
Tel: 86-592-2388138
Fax: 86-592-2388130

China - Zhuhai
Tel: 86-756-3210040
Fax: 86-756-3210049

India - Bangalore
Tel: 91-80-3090-4444
Fax: 91-80-3090-4433

India - New Delhi
Tel: 91-11-4160-8631
Fax: 91-11-4160-8632

India - Pune
Tel: 91-20-3019-1500

Japan - Osaka
Tel: 81-6-6152-7160
Fax: 81-6-6152-9310

Japan - Tokyo
Tel: 81-3-6880-3770
Fax: 81-3-6880-3771

Korea - Daegu
Tel: 82-53-744-4301
Fax: 82-53-744-4302

Korea - Seoul
Tel: 82-2-554-7200
Fax: 82-2-558-5932 or
82-2-558-5934

Malaysia - Kuala Lumpur
Tel: 60-3-6201-9857
Fax: 60-3-6201-9859

Malaysia - Penang
Tel: 60-4-227-8870
Fax: 60-4-227-4068

Philippines - Manila
Tel: 63-2-634-9065
Fax: 63-2-634-9069

Singapore
Tel: 65-6334-8870
Fax: 65-6334-8850

Taiwan - Hsin Chu
Tel: 886-3-5778-366
Fax: 886-3-5770-955

Taiwan - Kaohsiung
Tel: 886-7-213-7828

Taiwan - Taipei
Tel: 886-2-2508-8600
Fax: 886-2-2508-0102

Thailand - Bangkok
Tel: 66-2-694-1351
Fax: 66-2-694-1350

EUROPE
Austria - Wels
Tel: 43-7242-2244-39
Fax: 43-7242-2244-393

Denmark - Copenhagen
Tel: 45-4450-2828
Fax: 45-4485-2829

France - Paris
Tel: 33-1-69-53-63-20
Fax: 33-1-69-30-90-79

Germany - Dusseldorf
Tel: 49-2129-3766400

Germany - Karlsruhe
Tel: 49-721-625370

Germany - Munich
Tel: 49-89-627-144-0
Fax: 49-89-627-144-44

Italy - Milan
Tel: 39-0331-742611
Fax: 39-0331-466781

Italy - Venice
Tel: 39-049-7625286

Netherlands - Drunen
Tel: 31-416-690399
Fax: 31-416-690340

Poland - Warsaw
Tel: 48-22-3325737

Spain - Madrid
Tel: 34-91-708-08-90
Fax: 34-91-708-08-91

Sweden - Stockholm
Tel: 46-8-5090-4654

UK - Wokingham
Tel: 44-118-921-5800
Fax: 44-118-921-5820

07/14/15