AN2541

Functional Safety Demonstrator Using ATtiny3217

Features

- Core Independent Operation Using Configurable Custom Logic (CCL) and 16-Bit Timer/Counter Type A to Create Heartbeat Signal
- Core Independent Cyclic Redundancy Check Memory Scan (CRCSCAN)
- Core Independent Operation Using 12-Bit Timer/Counter Type D (TCD) to Drive a Fan Motor
- Core Independent TCD Fault Handling Using Event System (EVSYS), Analog Comparator (AC), and Digital-to-Analog Converter (DAC)
- Using Charlieplexing Technique to Drive a Large Number of LEDs with a Low Number of Pins, using 16-Bit Timer/Counter Type B (TCB) and Priority Interrupt
- Core Independent Watchdog Timer (WDT) in Window Mode
- Real-Time Counter Periodic Interrupt (RTC) (PIT)
- Board Controller with Touch Slider using the Peripheral Touch Controller (PTC) to Adjust Voltage to ATtiny3217, Demonstrating Voltage Level Monitor (VLM) interrupt, Brown-out Detector (BOD), and Power-on Reset (POR)
- Onboard Mini Embedded Debugger (mEDBG) for Programming and Debugging

Introduction

Safety and reliability are important and critical topics in many applications. Imagine if the buttons of a stovetop, that are controlled by a microcontroller, suddenly break. In this scenario, it is very important for the controller to detect the issue and safely turn the stovetop OFF to avoid potential danger. The safety and robustness features of the AVR® microcontrollers, in conjunction with the Class B firmware library, are valuable tools in handling robustness issues.

The Functional Safety Field Engagement Board demonstrates various features of an AVR microcontroller, such as Watchdog Timer (WDT), Cyclic Redundancy Check (CRC), Brown-out Detection (BOD), Voltage Level Monitoring (VLM), Power-on Reset (POR), and Timer/Counter type D (TCD) fault detection. In addition, Class B self-tests are designed to detect various faults on start-up or during program execution and shut down the application safely in case of a fault.

Table of Contents

Features ... 1

Introduction ... 1

1. Relevant Devices ... 4
   - 1.1. tinyAVR® 0-series ... 4
   - 1.2. tinyAVR® 1-series ... 4
   - 1.3. megaAVR® 0-series ... 5

2. Functional Safety Field Engagement Board ... 6
   - 2.1. Functional Safety Field Engagement Board Overview ... 6
     - 2.1.1. FSFEB Calibration ... 8
   - 2.2. Operating Voltage ... 9
     - 2.2.1. Voltage Level Monitoring (VLM) ... 10
     - 2.2.2. Brown-out Detect (BOD) ... 11
     - 2.2.3. Power-on Reset (POR) ... 11
   - 2.3. Application Heartbeat Setup ... 11
     - 2.3.1. TCA0 Setup ... 12
     - 2.3.2. CCL Setup ... 12
   - 2.4. Reset Register and Class B Status ... 13
     - 2.4.1. Reset Register Status ... 13
     - 2.4.2. Class B ... 15
   - 2.5. Window Watchdog Timer (WDT) Setup ... 16
   - 2.6. Cyclic Redundancy Check (CRC) Setup ... 18
   - 2.7. Fault Detection Using Event System Setup ... 19
   - 2.8. Charlieplexing Status LEDs ... 22
     - 2.8.1. Charlieplexed LEDs Setup ... 23
   - 2.9. Embedded Debugger Implementation ... 23
   - 2.10. CRC Checksum Setup ... 23

3. Get Source Code from Atmel | START ... 25

4. Revision History ... 26

The Microchip Web Site ... 27

Customer Change Notification Service ... 27

Customer Support ... 27

Microchip Devices Code Protection Feature ... 27

Legal Notice ... 28

Trademarks ... 28

Quality Management System Certified by DNV.....29

Worldwide Sales and Service.....30

1. Relevant Devices

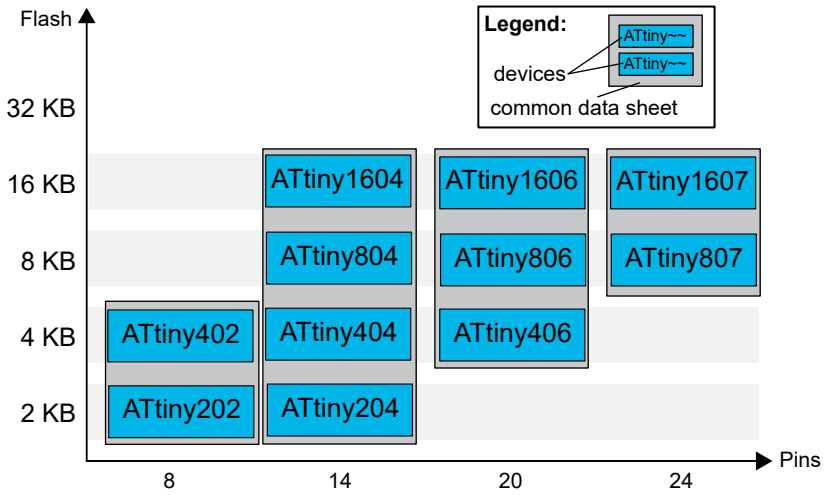
This chapter lists the relevant devices for this document.

1.1 tinyAVR® 0-series

The figure below shows the tinyAVR® 0-series, laying out pin count variants and memory sizes:

- Vertical migration is possible without code modification, as these devices are fully pin- and feature compatible.
- Horizontal migration to the left reduces the pin count and therefore, the available features.

Figure 1-1. tinyAVR® 0-series Overview



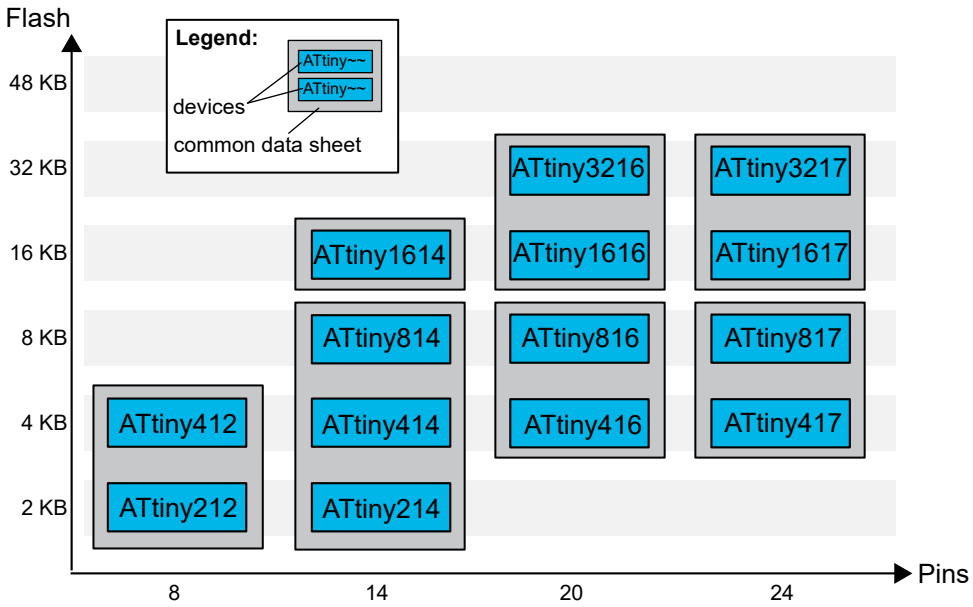
Devices with different Flash memory size typically also have different SRAM and EEPROM.

1.2 tinyAVR® 1-series

The figure below shows the tinyAVR® 1-series devices, laying out pin count variants and memory sizes:

- Vertical migration upwards is possible without code modification, as these devices are pin compatible and provide the same or more features. Downward migration may require code modification due to fewer available instances of some peripherals.
- Horizontal migration to the left reduces the pin count and therefore, the available features.

Figure 1-2. tinyAVR® 1-series Overview



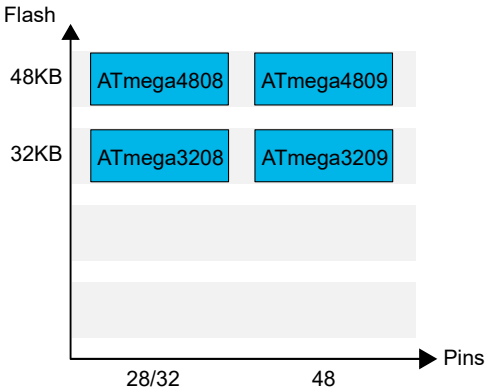
Devices with different Flash memory size typically also have different SRAM and EEPROM.

1.3 megaAVR® 0-series

The figure below shows the megaAVR® 0-series devices, laying out pin count variants and memory sizes:

- Vertical migration is possible without code modification, as these devices are fully pin and feature compatible.
- Horizontal migration to the left reduces the pin count and therefore, the available features.

Figure 1-3. megaAVR® 0-series Overview



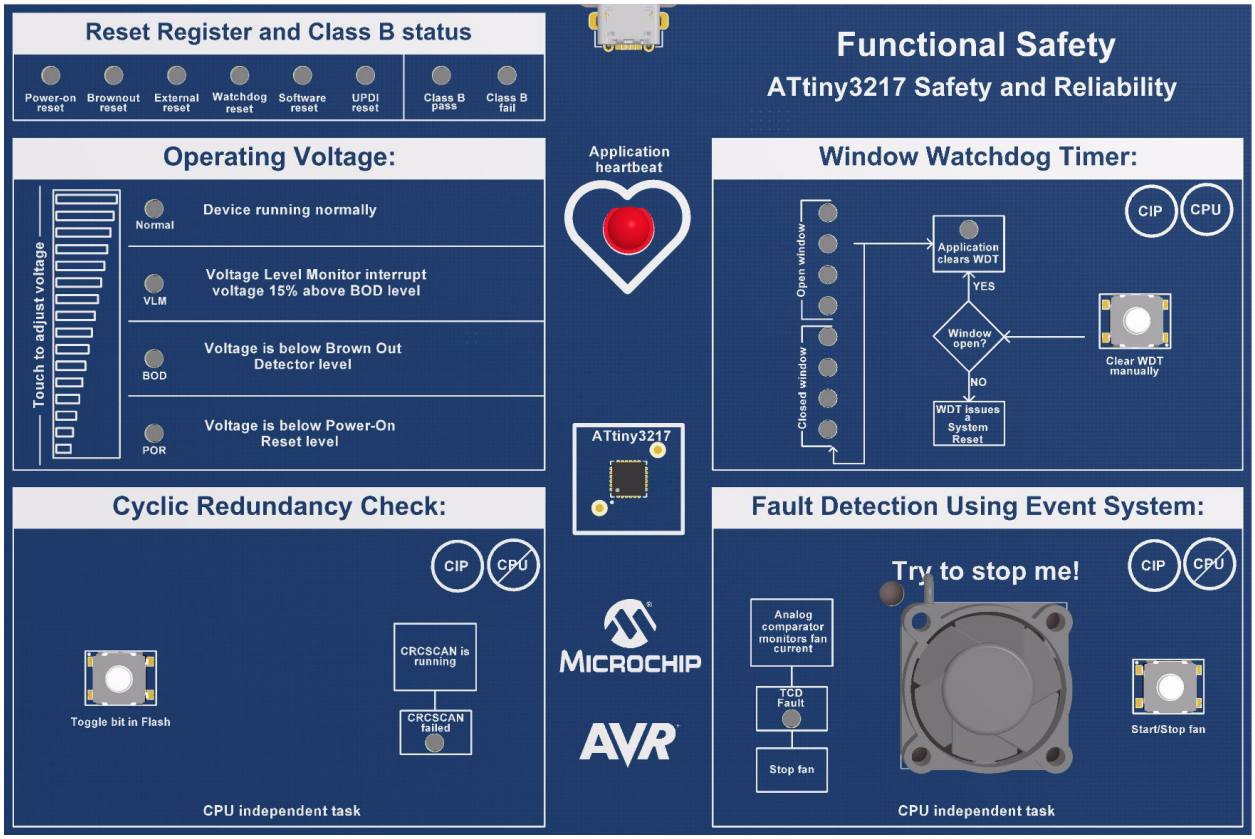
Devices with different Flash memory size typically also have different SRAM and EEPROM.

2. Functional Safety Field Engagement Board

2.1 Functional Safety Field Engagement Board Overview

The Functional Safety Field Engagement Board (FSFEB) demonstrates some of the safety and reliability features available on the tinyAVR 1-series devices. FSFEB is divided into five main sections, and each section has LEDs that show the current status to the user. All status LEDs, except for the four LEDs on the Operating Voltage section, are connected to ATtiny3217, and they are updated using Charlieplexing. A separate application heartbeat LED is showing that ATtiny3217 is operational and running the application. In addition, the application is running Class B code to test various systems in order to detect failures during start-up and run-time. To control the operating voltage of the board, and the corresponding LEDs, FSFEB also features a board controller MCU. This MCU is an ATtiny1617.

Figure 2-1. Functional Safety Field Engagement Board (FSFEB)



The following features are shown on FSFEB:

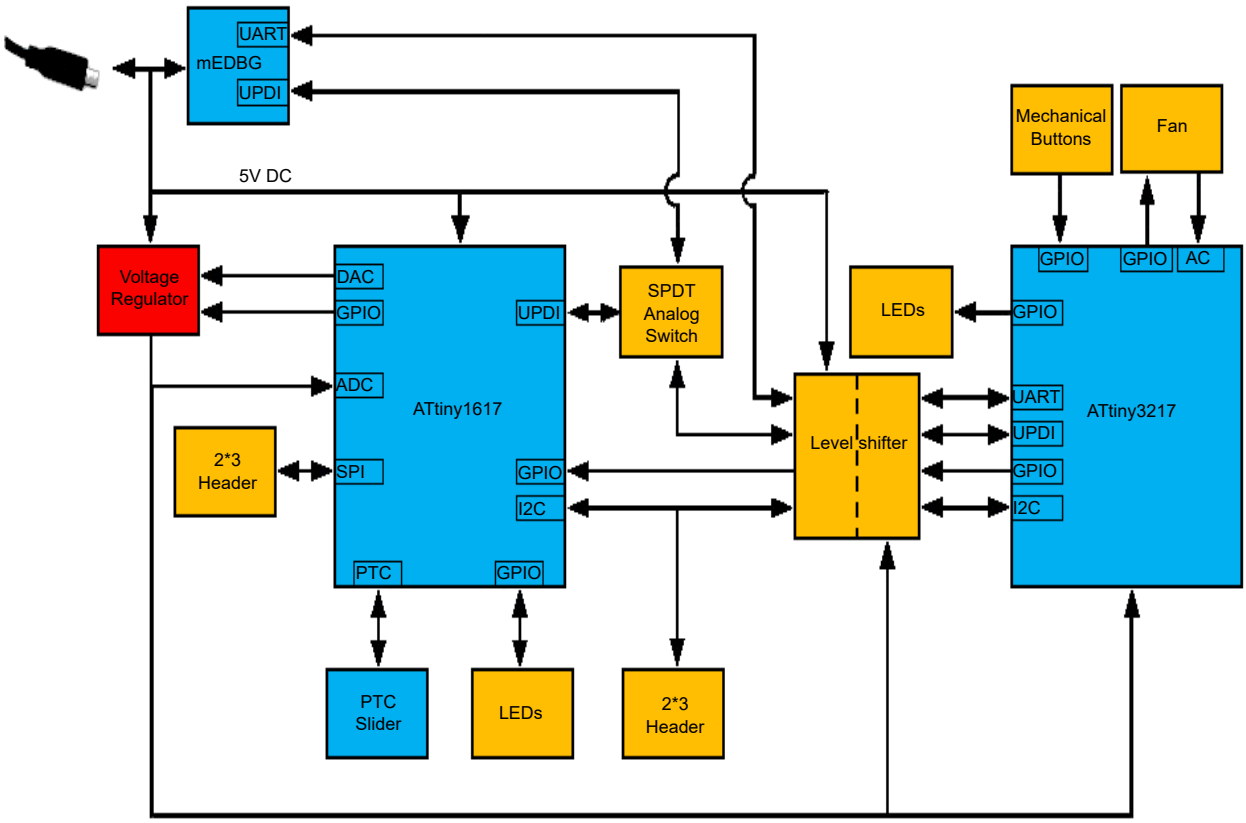
- Watchdog Timer (WDT) in window mode
- Cyclic Redundancy Check (CRC)
- Fault detection using Event system
- Voltage Level Monitor (VLM)
- Brown-Out Detect (BOD)
- Power On Reset (POR)
- Priority interrupt

The following support features are needed on FSFEB:

- Operating Voltage board controller
- mini Embedded Debugger (mEDBG)

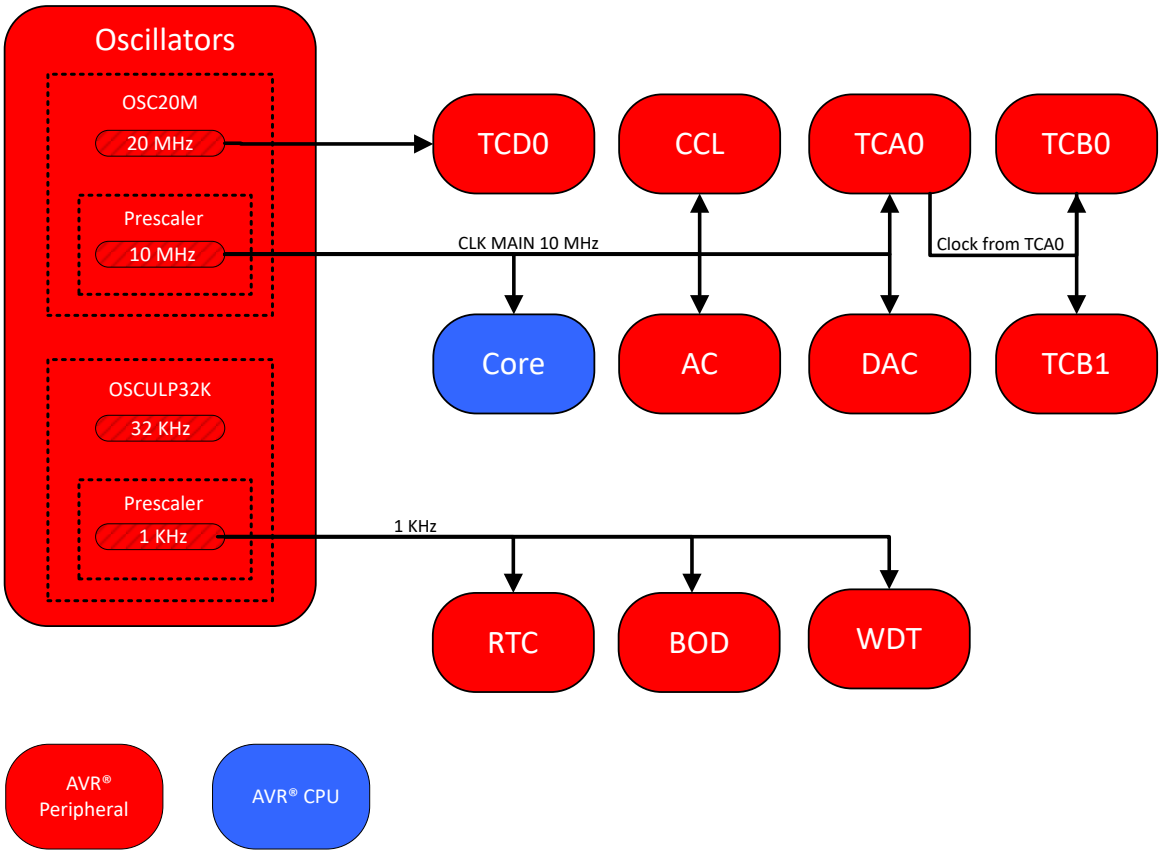
Figure 2-2 shows a block diagram of FSFEB and how ATtiny3217 is connected to the mEDBG and the board controller.

Figure 2-2. Functional Safety Block Diagram



The core and various peripherals can run on a variety of clocks. Figure 2-3 shows how the system is clocked.

Figure 2-3. Clock System Overview



2.1.1 FSFEB Calibration

It is highly recommended that FSFEB is calibrated on the first start-up, or if the behavior of FSFEB is not as expected.

The calibration routine will adjust the trigger level for the fault detection on the fan. Calibration is needed since the fan current consumption can vary based on temperature, supply voltage, and from device to device.

The calibration routine takes about eight seconds and can be started like this:

- Set operating voltage to POR
- Push and hold all three buttons
- Set operating voltage to Normal
- Release buttons

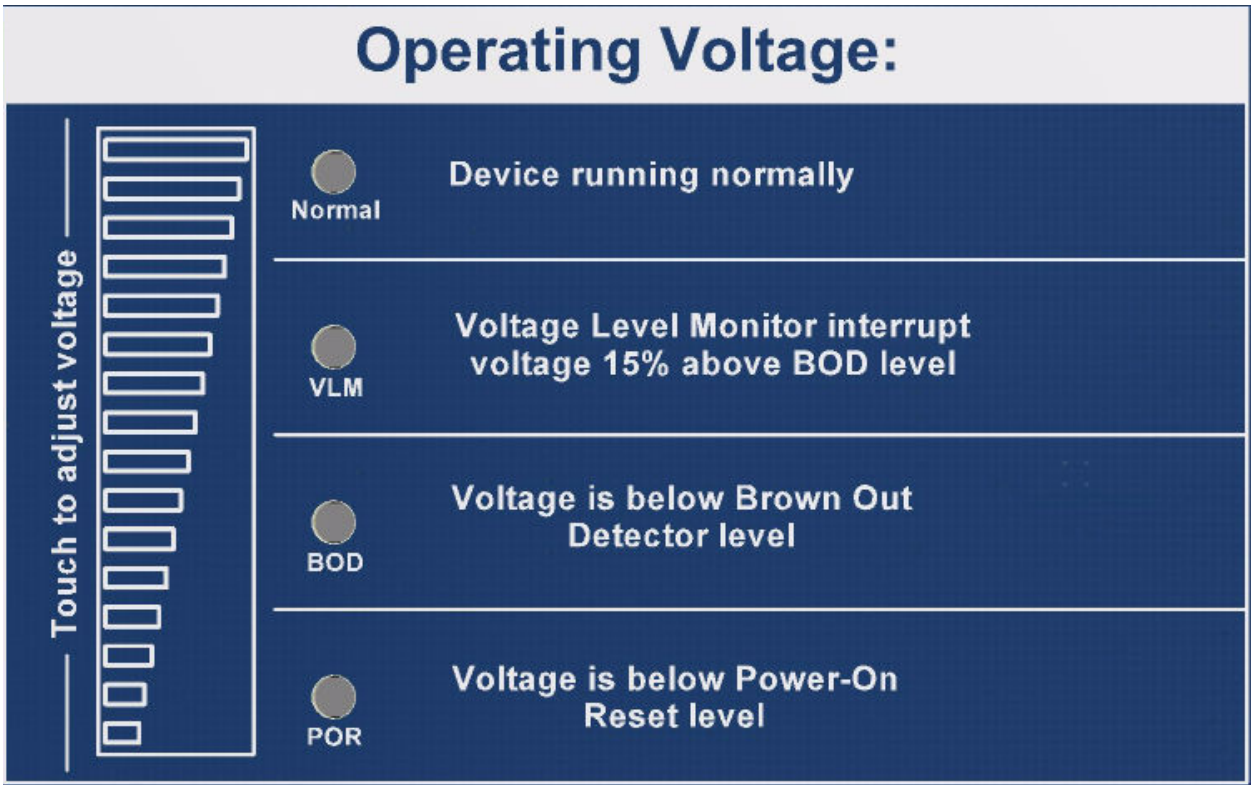
When FSFEB enters calibration mode, the Application heartbeat LED is turned ON (without blinking) and the fan will start. When calibration is complete, the calibration data is stored in EEPROM. These values will be used the next time ATtiny3217 is restarted.

A software reset occurs when calibration is complete, thus forcing a load of the new calibration data.

2.2 Operating Voltage

To demonstrate features like POR, BOD, and VLM in a realistic manner, a board controller is necessary to control the operating voltage of the ATtiny3217 device. Based on user input, implemented using a touch slider, the board controller will regulate the voltage supplied to ATtiny3217. With this setup, the user can adjust the slider and see the functionality of the POR, BOD, and VLM features and see the respective LEDs get set or cleared in the Reset Register, shown in Figure 2-1, based on what type of reset the device experienced on the previous reset.

Figure 2-4. Operating Voltage Overview



The board controller drives the status LEDs to indicate the voltage level supplied to ATtiny3217.

The different states are the following:

Normal

The operating voltage is set to approx. 5V and the Normal LED is set. The application heartbeat is stable at 60 beats per minute (bpm), as long as there are no other errors.

Voltage Level Monitor (VLM)

The operating voltage is set slightly (<15%) above the BOD level of ATtiny3217, and the VLM LED is set. The application heartbeat will increase to 90 bpm to indicate to the user that the VLM interrupt has triggered.

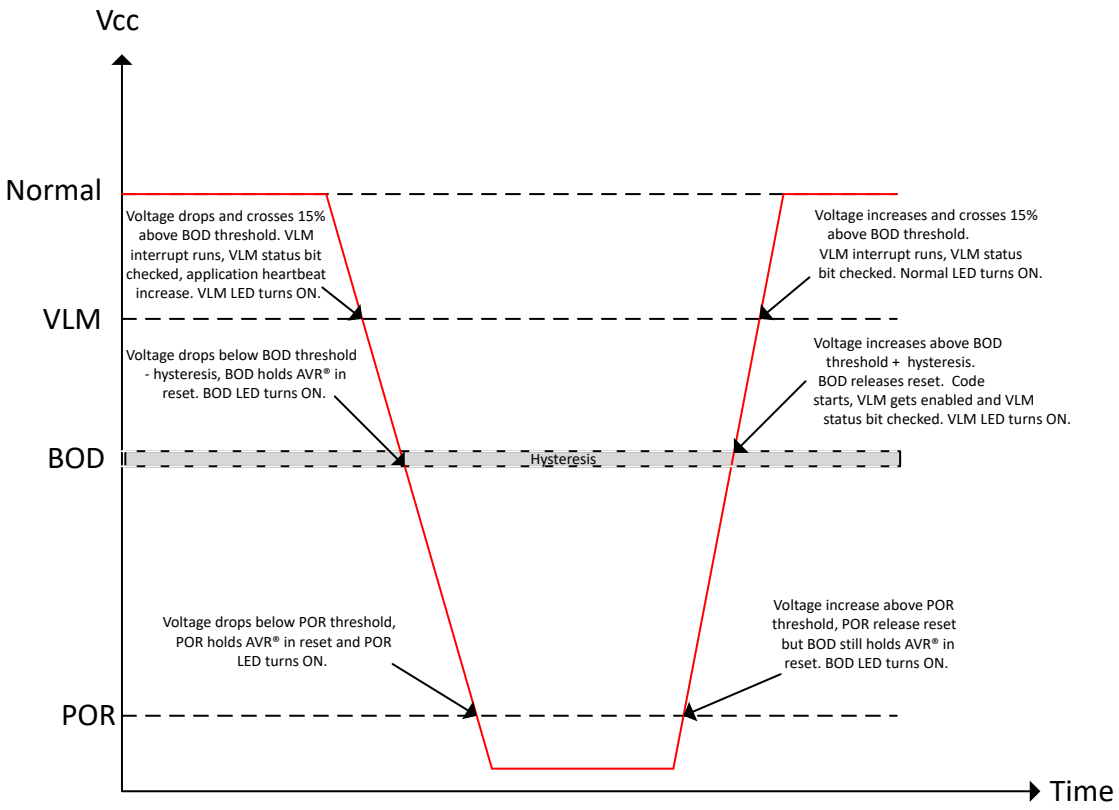
Brown-out Detect (BOD)

The operating voltage is set below the BOD level of ATtiny3217, and the BOD LED is turned ON. As the supplied voltage is below the BOD level, the BOD holds ATtiny3217 in reset. The application is not running and the application heartbeat is off.

Power-on Reset (POR)

The operating voltage is set below the POR threshold of ATtiny3217, and the POR LED is turned ON. As the supplied voltage is below the POR level, the POR holds ATtiny3217 in reset. The application is not running and the application heartbeat is off.

Figure 2-5. Operating Voltage Diagram



2.2.1 Voltage Level Monitoring (VLM)

The VLM, when enabled, monitors the power supply and can be configured to generate an interrupt request when the voltage passes a given threshold. This can act as an early warning to the application that the voltage is passing the VLM threshold and the application can take necessary action to prepare for a possible Brown-out or start-up. The threshold is configurable and is expressed in percentage above the configured BOD level.

The VLM threshold can be configured to:

- 5% above BOD threshold
- 15% above BOD threshold
- 25% above BOD threshold

The VLM can be configured to trigger an interrupt request when:

- Voltage crosses VLM threshold from above
- Voltage crosses VLM threshold from below
- Voltage crosses VLM threshold from either direction

On FSFEB, the VLM is configured to trigger an interrupt request when the voltage crosses in either direction 15% above the BOD threshold. FSFEB will always start on 0V after a power-up, and when the user sets the voltage above BOD level using the touch slider, the device will start. As soon as the VLM is enabled, the VLM status bit will be set if the voltage is below the VLM threshold. The VLM interrupt will only trigger when the voltage crosses the VLM threshold.

In the VLM interrupt routine, the VLMS bit in the VLM status register is read to determine if the voltage is above or below the VLM threshold. If the VLMS bit is one, the operating voltage is below the threshold and the application heartbeat is increased to indicate this. If the VLMS bit is zero, the operating voltage is above the threshold and the application heartbeat is set to 60 bpm, indicating normal operation.

2.2.2 Brown-out Detect (BOD)

Information regarding BOD reset and other reset sources can be found in Reset Register Status.

On FSFEB, the BOD threshold is set to 2.87V. This means that when the voltage goes below 2.87V, the BOD will hold the device in reset until the voltage is above 2.87V again.

The BOD LED is connected to the board controller. Since ATtiny3217 is in BOD reset when this LED should be ON, it cannot be controlled by ATtiny3217 itself.

2.2.3 Power-on Reset (POR)

Information regarding POR reset and other reset sources can be found in Reset Register Status.

When the user sets the voltage on the slider to POR level, the POR will take control and hold the device in reset until the voltage rises again above the threshold.

The POR LED is connected to the board controller. Since ATtiny3217 is in POR reset when this LED should be ON, it cannot be controlled by ATtiny3217 itself.

2.3 Application Heartbeat Setup

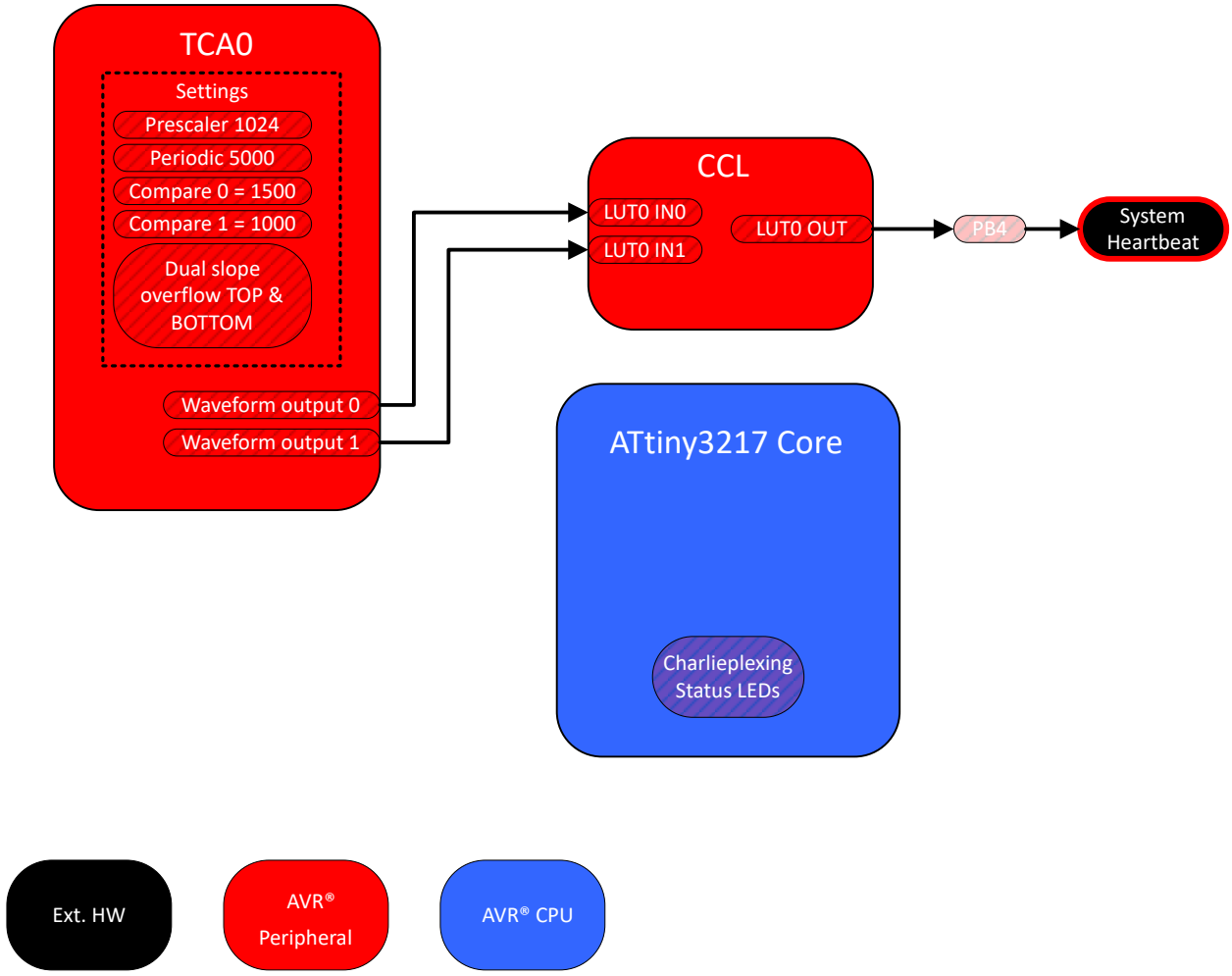
The application heartbeat is a visual way to show the user whether the application is running or not. When the application is in a stable operating state, a heartbeat with a frequency of 60 bpm is displayed. When a problem is detected by the device, the heartbeat is accelerated and the correct status LED is set, indicating what the failure is.

As shown in Figure 2-6, the heartbeat LED is driven by TCA0, feeding two waveforms with different frequencies into to the CCL LUT0. The CCL will mix together the PWM signals to create a realistic heartbeat. The heartbeat is running core independently, and only if a change to the bpm is needed will the CPU update the TCA0 period buffer registers. The heartbeat will then continue with the new bpm configuration, without any need for further CPU execution.

The following will result in an increased bpm:

- CRC error detected
- Class B self-tests fail
- Fault detection triggered
- VLM interrupt triggered, either from voltage rising from BOD to VLM or falling from Normal to VLM

Figure 2-6. Application Heartbeat Overview



2.3.1 TCA0 Setup

TCA0 is a 16-bit timer/counter with three compare channels and six PWM outputs. In order to create a realistic application heartbeat, two PWM signals are needed, generated using two compare channels.

To create a heartbeat of approximately 60 bpm, the timer setup needs to be as follows:

- Divide system clock by 1024
- Set period register to 5000
- Enable compare channel 0 and set the compare value to 1500
- Enable compare channel 1 and set the compare value to 1000
- Enable compare waveform output for both channels
- Set waveform generation mode to; Dual-slope PWM, overflow on TOP and BOTTOM

When the application needs to change the heart rate, the CPU will change the value in the period buffer register.

2.3.2 CCL Setup

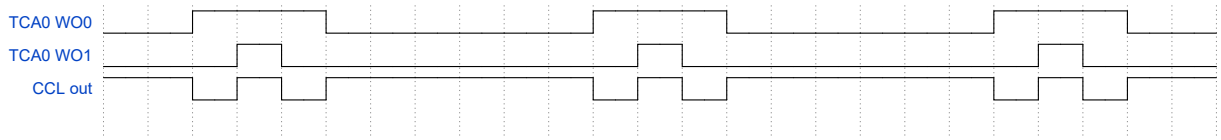
The CCL is a programmable logic peripheral which can be connected to pins, events, or other internal peripherals. The CCL can serve as glue logic between the device peripherals and external devices.

On the functional safety demonstrator the LUT0 inputs are connected like this:

- IN[0] connected to TCA0 wave output 0
- IN[1] connected to TCA0 wave output 1
- IN[2] is masked

The CCL is configured as an exclusive NOR gate, which means that when the input signals to the CCL are not equal, the CCL output will become "0".

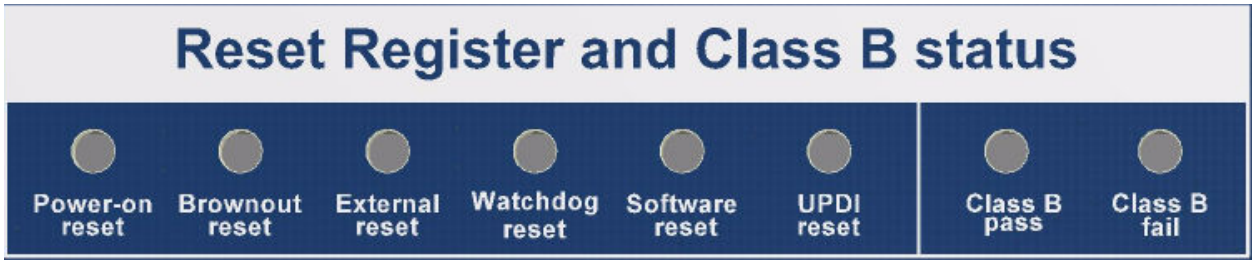
Figure 2-7. Application Heartbeat CCL Waveform



2.4 Reset Register and Class B Status

Reset register and Class B status LEDs are a part of the charlieplexed LED matrix. They show the cause of the last reset of the device and current Class B self-test status. The reset status holds useful information for Class B self-tests and other self-diagnostic software.

Figure 2-8. Reset Register and Class B Status



2.4.1 Reset Register Status

The Reset Flag register (RSTCTRL.RSTFR) on the AVR will contain the reset source of the last reset. The Reset Register status LEDs display the content of the Reset Flag register inside the AVR. Reading out the reset cause makes it possible for the application code to log and report errors and unstable behavior that caused the device to be reset.

On ATtiny3217 there are six different sources that can reset the device, as shown in Figure 2-9.

- Power supply Reset sources:
  - Brown-out Detect (BOD)
  - Power-on Reset (POR)
- User Reset sources:
  - External Reset Pin
  - Watchdog Reset
  - Software Reset
  - UPDI Reset

Power-on Reset (POR)

During a power-up of a device, it is important to give the entire device a reset to put everything in a known state. It is equally important not to start executing code from Flash before the Flash and digital logic have sufficient power. When the voltage rises, the POR is activated and will hold the device in reset

until the voltage is above a fixed threshold value. The POR will remain enabled as long as the device is powered.

Brown-out Detect (BOD)

The BOD monitors the power supply and compares the voltage against two threshold levels. The BOD is used on systems where the voltage should not drop below a certain level due to, e.g., the speed the device is running at or external hardware that requires a certain voltage level. If the voltage falls below the configured threshold, the BOD issues a system reset and will hold the device in reset until the voltage has risen above the threshold.

External Reset Pin

The External Reset pin is a port pin that, if enabled, will hold the device in reset when held low. The external reset will reset the entire device, except for the UPDI and TCD pin override settings.

External reset is not demonstrated on FSFEB since the UPDI and Reset shares the same pin. FSFEB uses the UPDI functionality for programming and debugging of the device.

Watchdog Reset

The Watchdog Timer (WDT) is a peripheral to help ensure correct program operation. It will recover the device from situations such as runaway or deadlocked code, by issuing a reset. When enabled, the WDT is a constantly running timer, configured to a predefined timeout period. If the WDT is not cleared within the timeout period, the WDT will issue a system reset.

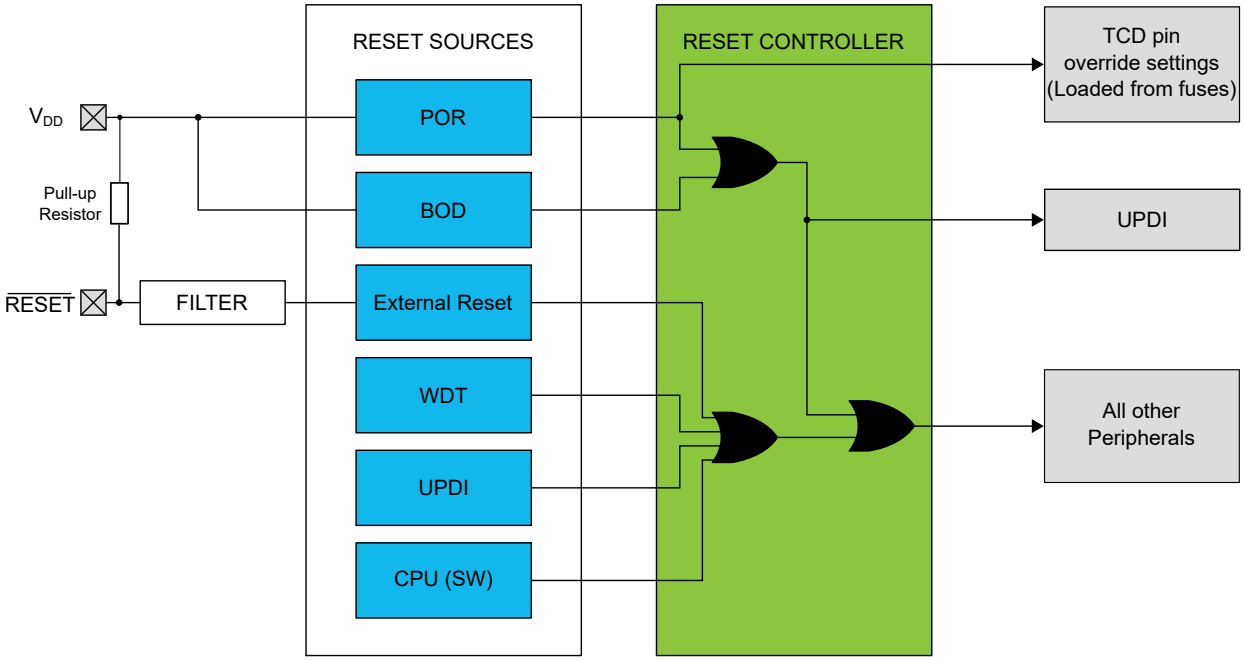
Universal Program Debug Interface (UPDI) Reset

The UPDI contains a separate reset source that is used to reset the device during external programming and debugging. The reset source is accessible only from external debuggers and programmers. The UPDI reset will reset all logic except the UPDI itself, TCD pin override settings, and BOD configuration.

Software Reset

The software reset makes it possible to issue a system reset from software. The reset is generated by writing a “1” to the Software Reset Enable bit (SWRE) in the Software Reset register. The reset will take place immediately after the bit is written, and the device will be kept in reset until the reset sequence is completed. All logic is reset on software reset, with the exception of UPDI, TCD pin override settings and BOD configuration.

Figure 2-9. Reset System Overview



2.4.2 Class B

Modern appliances are mostly electronically controlled. Electronic controls enable higher efficiencies, additional functionality, and an improved user experience. But what happens if something goes wrong? IEC 60730 addresses the safety of electronic controls in appliances. This standard is also referred to by other standards for safety-critical devices, for example, IEC 60335. Currently, IEC 60730 is mandatory for appliances sold in Europe, and widespread adoption is likely.

IEC 60730 Annex H defines three classes of control software for appliances:

- Class A - Control functions which are not intended to be relied upon for safety of the equipment.
- Class B - Software that includes code intended to prevent hazards if a fault, other than a software fault, occurs in the appliance.
- Class C - Software that includes code intended to prevent hazards without the use of other protective devices.

For an appliance to comply with the Class B requirements, the control software must detect and handle faults for system components. For a customer wishing to certify a product in accordance with Class B requirements, Microchip offers a Class B library for tinyAVR® 1-series. This library implements the tests needed to be in compliance with Class B and can reduce development time and cost for the customer.

Tests are done on CPU registers, program counter, frequency, CRC, interrupt handling and execution, clock, SRAM, Flash, and EEPROM, and peripherals such as ADC, DAC, and WDT.

For more information, refer to *Guide to IEC 60730 Class B Compliance with tinyAVR 1-series*.

2.4.2.1 Class B Self-Tests

The various Class B self-tests implemented on FSFEB are:

- SRAM test
- CRC test
- Button test

SRAM Test

On start-up, the device will run a test of the internal SRAM. The purpose of the test is to detect stuck bits and coupling faults in SRAM and on the data bus, and any addressing problems. The internal SRAM is used for volatile storage of data, and any faults related to this can cause issues for the appliance control.

CRC Test

On start-up, the device will run a CRC test to determine if Flash content is unchanged, by calculating a checksum and comparing this to a checksum programmed at the end of Flash. The CRC test will also be run once every second after start-up. The purpose of the test is to detect changes in Flash as soon as they happen and take action to stop the device from executing potentially wrong Flash content.

Button Test

On start-up, the device will check all three buttons on FSFEB if they are pressed down or not. If a button is pressed down, the class B self-test fail LED will be turned ON. The purpose of the test is to determine if the buttons are broken or not.
Note: If all three buttons are pressed down during start-up, FSFEB will enter calibration mode.

To be able to get the Class B pass LED to turn ON, all Class B self-tests must pass. If any of the tests fail, the class B fail LED will turn ON.

2.5 Window Watchdog Timer (WDT) Setup

Information regarding WDT reset and other reset sources can be found in Reset Register Status.

The WDT on ATtiny3217 has two modes of operation; Normal mode and Window mode. FSFEB is using the Window mode operation. As shown in Figure 2-10, when using Window mode, the timeout period is split into two parts; the closed window and the open window. While in the closed window, attempting to clear the watchdog will result in a system reset. This is also reflected in the markings on FSFEB, as shown in Figure 2-11.

Figure 2-10. Window WDT

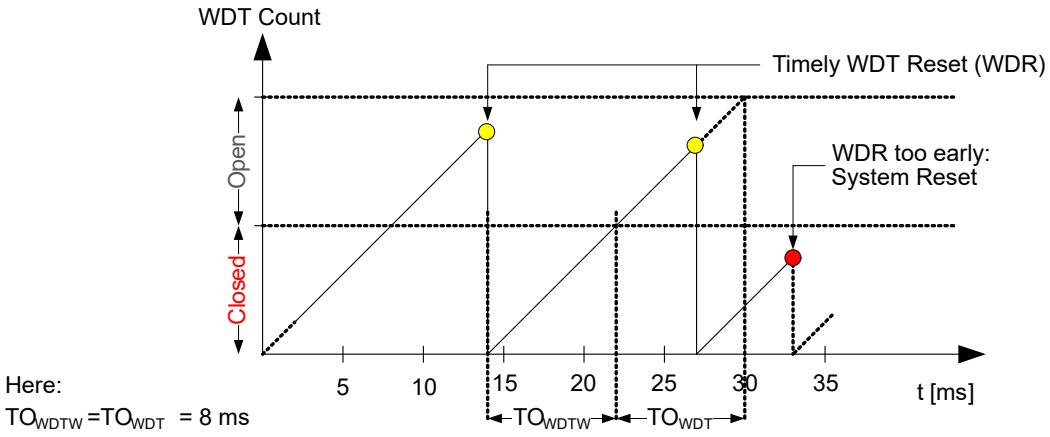
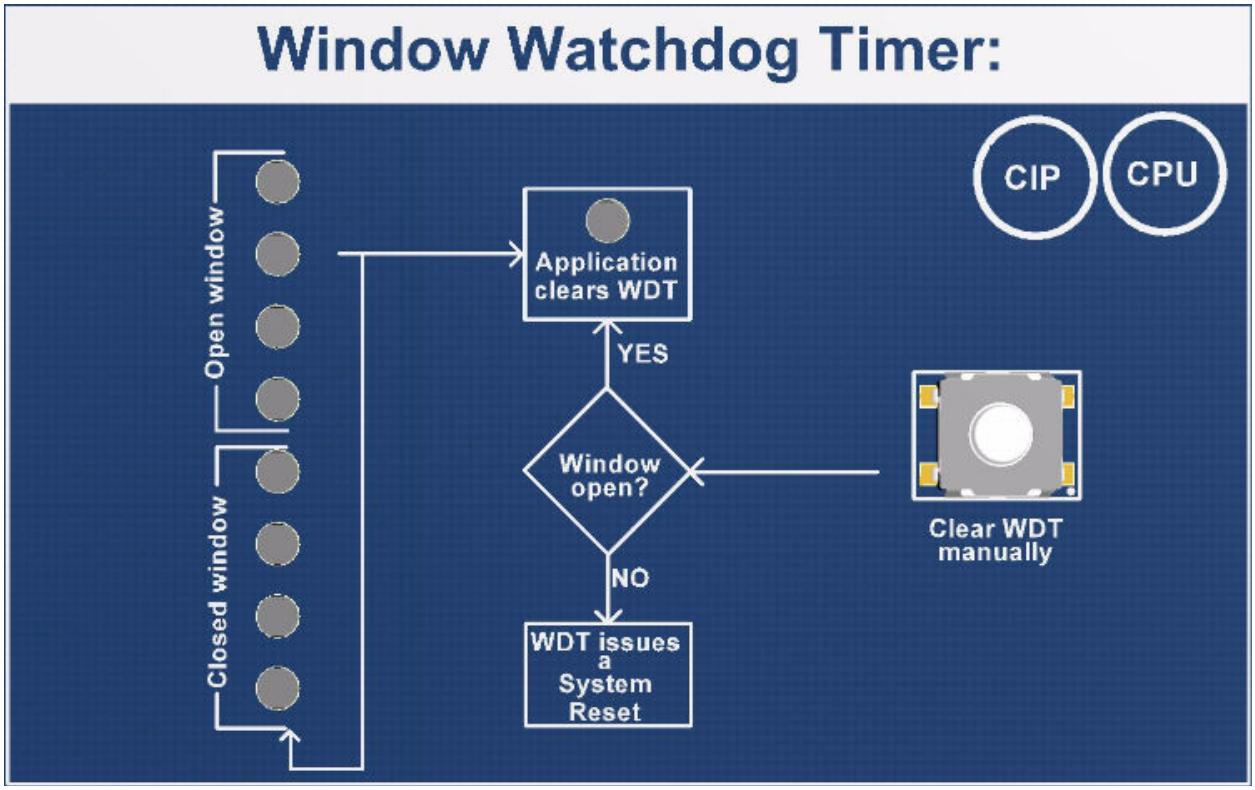


Figure 2-11. Window Watchdog Timer



The WDT Window mode operation is configured as shown below:

- Timeout = 8.2 seconds
- Closed Window = 4.1 seconds
- Open Window = 4.1 seconds

The RTC is configured to give a periodic interrupt each second. This is used to update a variable that counts the number of seconds since the last time the WDT was cleared. The RTC and WDT are running on the same clock, which means that the RTC periodic interrupt and the increase of the WDT counter will happen approximately at the same time.

The charlieplexed LEDs illustrating the closed and open window will light up, one LED per second. When the counter reaches 7 seconds, which is within the open window, the CPU will run the WDR instruction that clears the WDT counter. The application code will blink the WDT LED to indicate that the CPU has cleared the WDT and the counting will restart from zero. As long as the application keeps clearing the WDT reset flag inside the open window, the WDT will never reset the device.

A button is added to make it possible to manually clear the WDT. This means that the user can reset the device in the closed window on purpose in order to see the WDT reset the device. It is also possible to manually clear the WDT in the open window by using the button. This will not reset the device.

Note: When using the Cyclic Redundancy Check and an error is inserted in the Flash, the application code is written in such a way that it will stop clearing the WDT counter, and the WDT will get a timeout after 8.2 seconds and the device will be reset. After the reset, the error inserted in Flash will be written back to the original value by the application code. Refer to the Cyclic Redundancy Check section for more details.

2.6 Cyclic Redundancy Check (CRC) Setup

A Cyclic Redundancy Check (CRC) takes a data stream of bytes from the NVM and generates a checksum. The CRCSCAN peripheral can be used to detect errors in program memory.

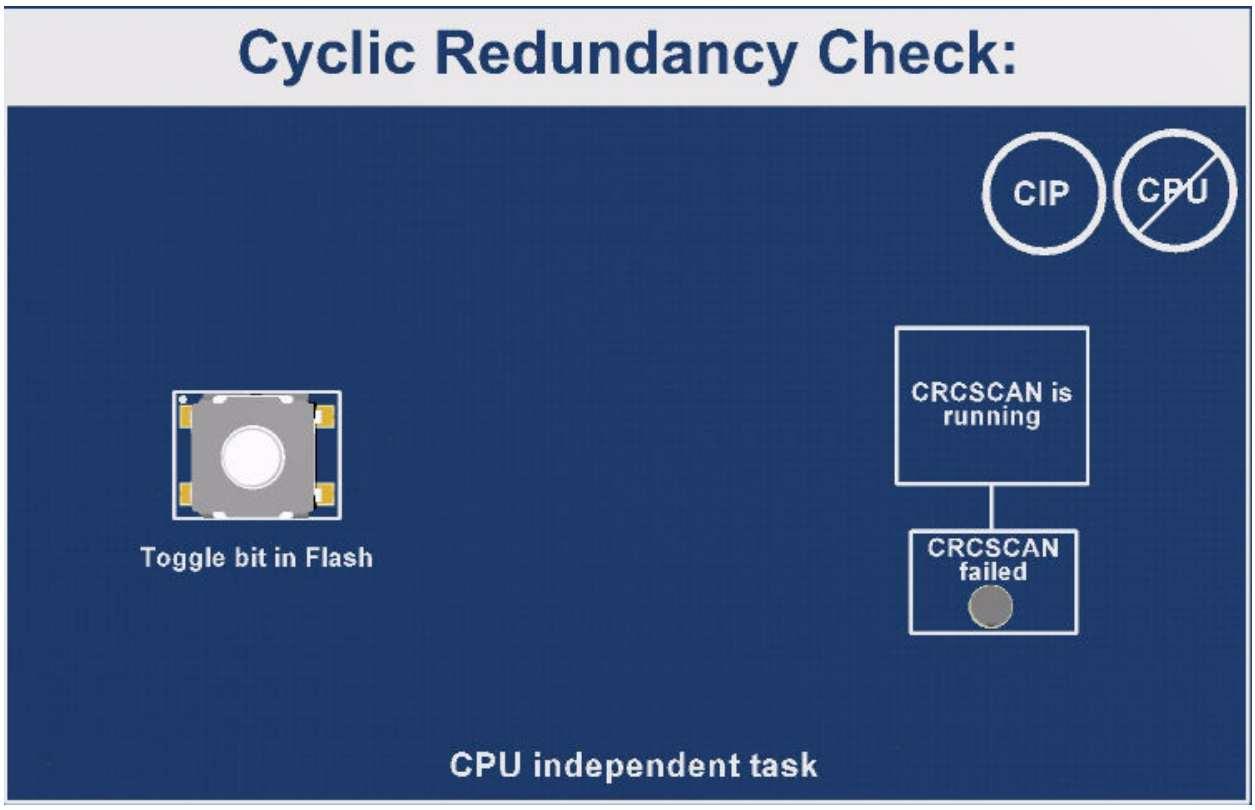
The CRCSCAN peripheral expects a two byte CRC checksum to be located at the end of the area selected for scanning. For the FSFEB project, this means that a checksum has to be calculated and added to the image before it is programmed onto the device. For details on how to do this, refer to CRC Checksum Setup.

A property of the CRC module is that when a scan is started, the CRC module has priority access to the Flash. This means that the CPU is stalled until the CRC module is done. If the CRC scan passes, the OK bit in the CRC status register is set to '1'. If the scan fails, the bit is set to '0'.

In the application code, the CRC scan will be started each time the RTC gets a periodic interrupt. This happens once every second.

To be able to demonstrate the function on the CRC, it is possible to insert an error in Flash by using the *Toggle bit in Flash* button, as shown in Figure 2-12. When the button is pressed, the CPU will write to an unused area of Flash within the defined CRC section. When the next CRC scan is started, the CRC will get a different checksum and set the OK bit in the CRC status register to '0'. The CPU checks this bit regularly and when it detects that this bit is '0', the CRC has failed. The CRCSCAN and Class B fail LEDs are turned ON and the Class B pass LED is turned OFF, and the application heartbeat frequency is increased to signal the user that the CRC scan has failed.

Figure 2-12. Cyclic Redundancy Check



When the CRC scan fails, the application code will no longer clear the WDT. This means the device will be reset when the WDT times out. After the reset, the CPU will write back the correct value to the Flash address that was written to during the CRC test, to make FSFEB ready for new tests.

The CRC has the possibility to trigger a CRC interrupt when failure is detected. This feature should be selected if the user wants to make sure that the application is stopped when errors are detected in Flash. This feature is not used on FSFEB. As the CRC interrupt cannot be cleared, the device would become stuck in the interrupt handler. Due to the handling of the charlieplexed LEDs, it would become impossible to set the CRC failure LED, and FSFEB would appear to be locked.

The fact that the CRC interrupt cannot be cleared once an error has been detected is a security feature. When the CRC has detected a Flash failure, which can be anywhere in the Flash, no code is safe to execute. It is recommended to prevent the device from executing code.

The timing of the CRC operation is decided by two factors: The system clock and the size of the area to be scanned. Every third clock cycle, a 16-bit word is read from Flash, so the time it takes to scan a section of Flash can be calculated by following this formula:

$$T_{SCAN} = \frac{B_{SCAN}}{2} / \frac{F_{CPU}}{3}$$

where

- T_{SCAN} is the time it takes to scan the area, in seconds
- B_{SCAN} is the size of the area to scan, in bytes
- F_{CPU} is the operating frequency of the CPU, in MHz

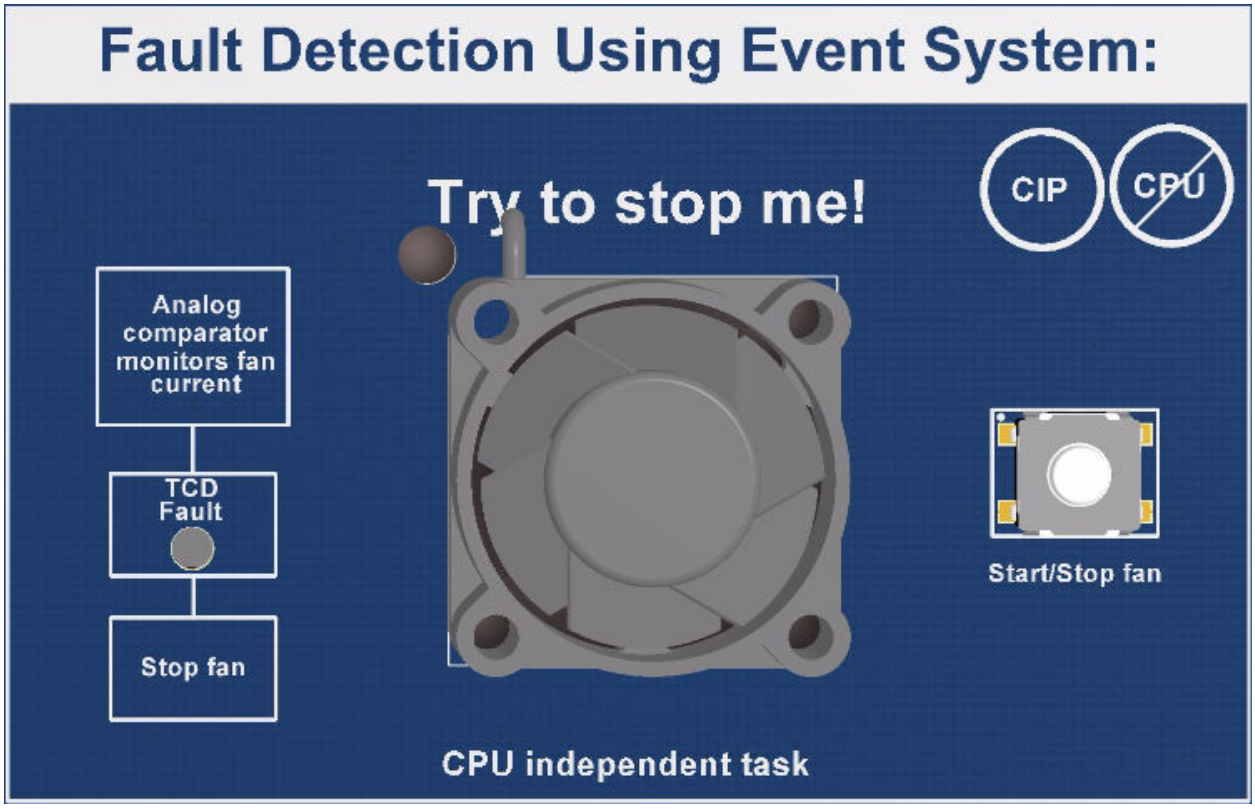
2.7 Fault Detection Using Event System Setup

The fault detection using Event System shows how the TCD, which drives a fan, can automatically stop the fan from a fault condition. The AC and DAC monitor the current consumption of the fan. If a current consumption above the normal level is detected, which will be considered as a fault condition, a signal is sent to the TCD. The signal is sent using the Event System, which is CPU independent. The TCD automatically stops the fan upon receiving the event.

The fan motor on FSFEB is driven by the TCD. Since this is not a motor control demonstration, but rather a demonstration of fault detection and automatic shutdown, the motor control part is done in a very simple way. Ideally, a stable DC voltage from TCD would be best to power the fan motor since it already has a speed controller onboard. However, as the TCD cannot provide this, the signal will be PWM with a very short low period.

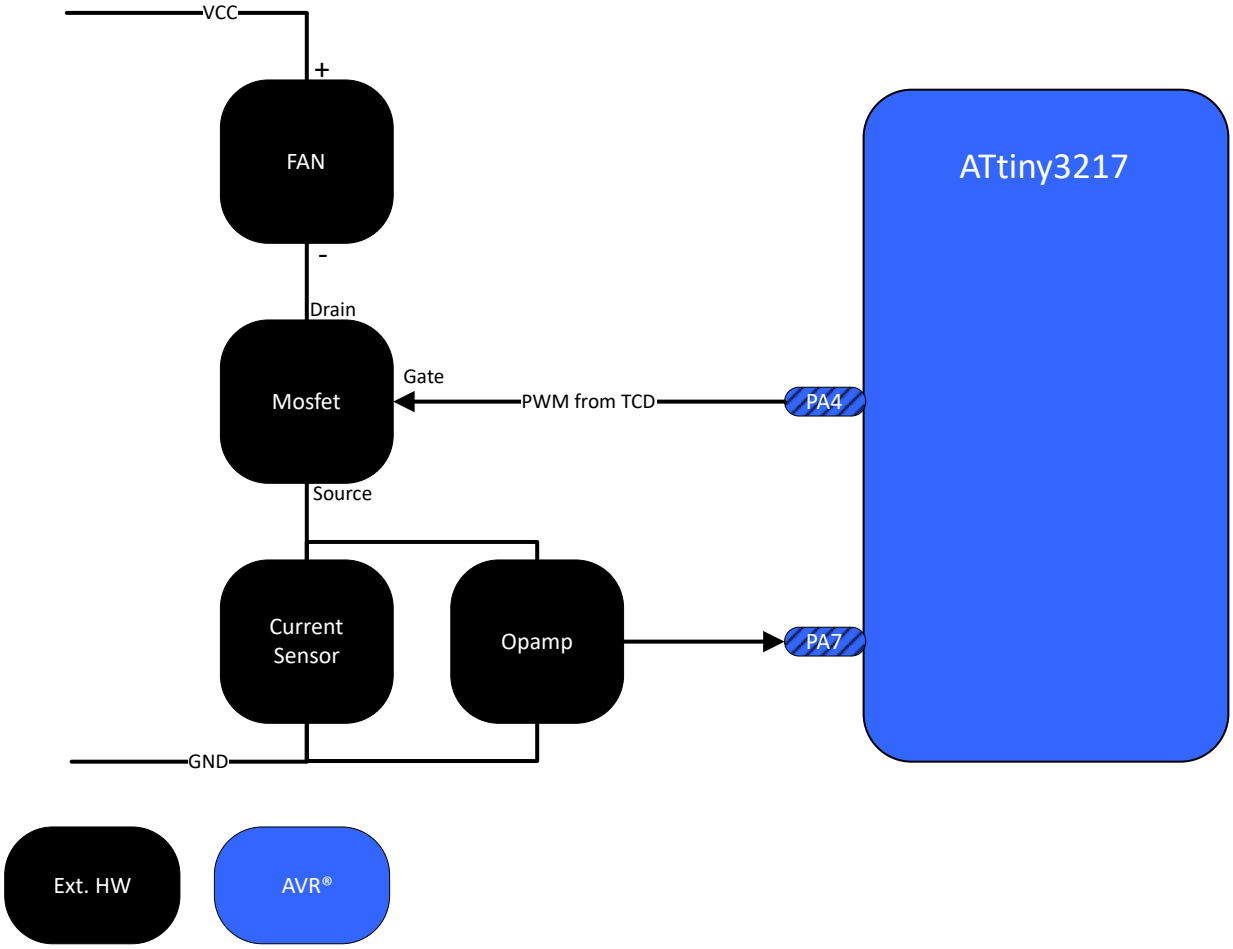
The fan will start when the user pushes the Start/Stop fan button, as shown in Figure 2-13. It will run until the button is pressed again, or the fan is loaded enough to trigger a fault. The idea is that the user can start the fan and then load or even stop the fan using a finger. The current consumption will increase past the set limit, the AC will send an event to the TCD and this will trigger the fault detection functionality and automatically shut down the fan.

Figure 2-13. Fault Detection Using Event System



To be able to measure the fans current consumption, a small resistor is needed in series with the fan motor and ground. The voltage drop over the resistor will be amplified through a simple op-amp circuit and connected to the positive input on AC. The measurement technique is called low-side current sensing. It is shown in Figure 2-14.

Figure 2-14. Fan Control and Low-Side Current Sensing



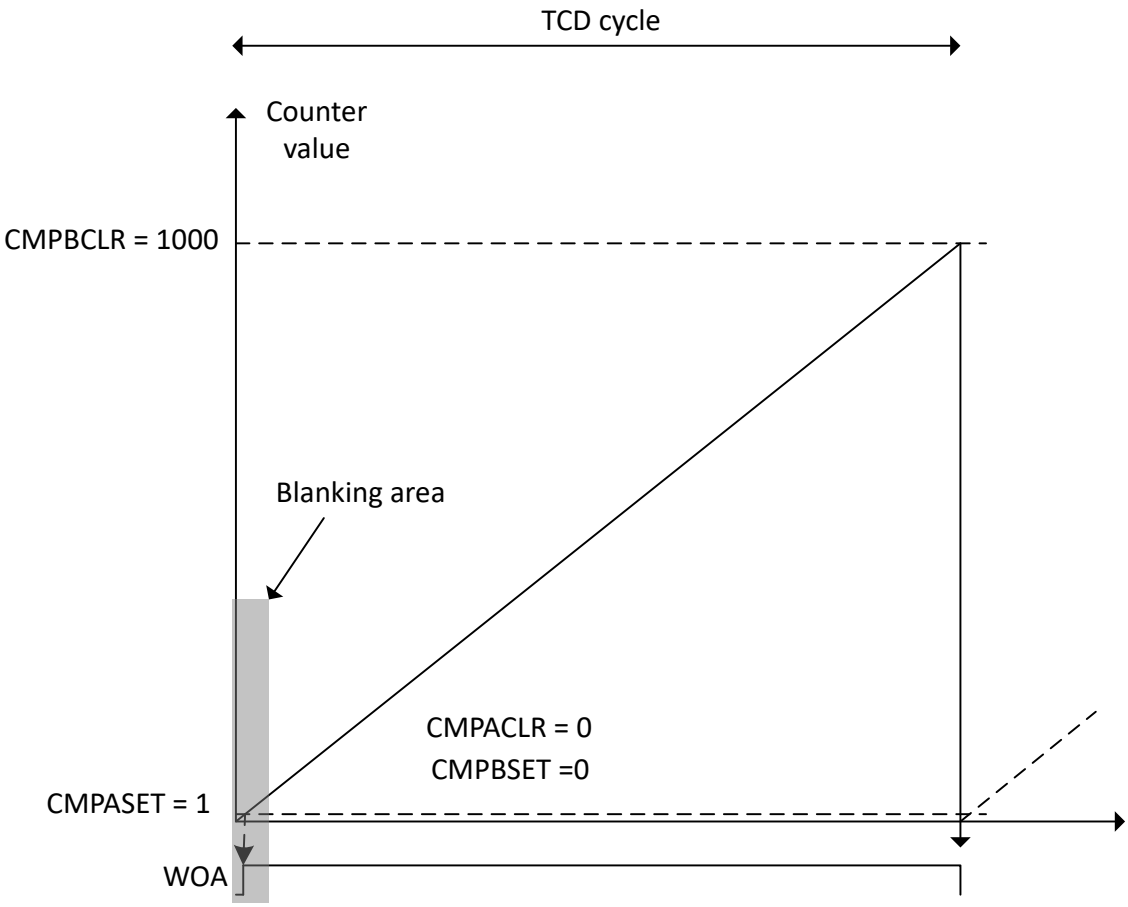
The internal DAC is connected to the negative input on the AC and is used to set the trigger level for the AC. The AC output is further connected to TCD0 input A through the event system. When the fan is started the current consumption is measured through the series resistor and the voltage drop is amplified through the op-amp. If the current consumption is lower than the level set by the DAC, the AC output will be low. As soon as the fan is loaded the current consumption will increase above the set DAC level and the AC output will become high. This will create an event to the TCD. The TCD will then stop driving the fan and wait until the user restarts the fan.

The current consumption is at its highest when the fan is started and decreases as the fan speed increases. To avoid start-up current from triggering the AC the TCB1 timer interrupt is used to gradually decrease the DAC value the first second after the button is pressed down. The first time the TCB1 interrupt triggers the DAC value is set high, and the TCD0 timer is started.

Sometimes, when the TCD PWM output signal goes from low to high, a spike in current generated by the motor inductance might falsely trigger the AC to generate an event that stops the timer. To avoid this, the timer is configured to use the input blanking functionality. The timer can be set up to mask out the inputs for a programmable time in the selected TCD cycle. This means that the timer will ignore any event if it happens within this time period.

When the fault detection is triggered, the TCD fault LED will be turned ON and the application heartbeat frequency is increased to signal to the user that something is wrong. If the Start/Stop button is not pressed within 3 seconds, the application code will execute a software reset to reset the device.

Figure 2-15. TCD Timing



2.8 Charlieplexing Status LEDs

All the LEDs connected to ATtiny3217, except the application heartbeat, are updated using the charlieplexing technique.

Charlieplexing is a technique that takes advantage of the tri-state logic capabilities of ATtiny3217 for driving a high number of LEDs using only a few pins. On FSFEB, this method was chosen since a large number of LEDs are necessary to display the status of all the modules, and most of the pins on ATtiny3217 were used for other things.

The technique is efficient in its use of I/O pins but can be complicated to use in large designs due to issues with duty cycle, current requirements, and LED forward voltage considerations.

Because only a single set of LEDs, with a common anode or cathode, can be lit simultaneously without turning unintended LEDs ON, charlieplexing requires frequent output changes, through a method known as flickering. Not all LEDs are lit simultaneously, but rather one set of LEDs are lit briefly, then another set, then another, and eventually the cycle repeats. If it is done fast enough, due to the human eye persistence of vision, the LEDs will appear to be ON all the time.

On FSFEB there are 19 charlieplexed LEDs, using only five I/O pins. The update frequency is set to 60 Hz.

The LEDs for the following modules are charlieplexed:

- Reset register and class B status
- Window Watchdog Timer
- Cyclic Redundancy Check
- Fault detection using Event System

More information about charlieplexing can be found here.

2.8.1 Charlieplexed LEDs Setup

The update frequency must be above 50 Hz to avoid visible flickering of the LEDs. This means that the CPU needs to prioritize this task before all other tasks to get a systematic and stable LED update frequency. To achieve this the TCB0 timer is set up to use the clock from TCA0 and configured to give periodic interrupt with a frequency of 60 Hz. In addition, the TCB0 interrupt is set up to be priority level 1 interrupt to make sure that nothing can disturb the update.

Only one interrupt can be configured to level 1 at one time so the other interrupts have level 0. This means that this interrupt is always executed before other interrupts and if the device is busy servicing a different interrupt. When level 1 interrupt is triggered, the ongoing interrupt is halted, and the level 1 interrupt is executed. When this interrupt is done the core will resume the halted interrupt.

Each LED has an assigned byte in a char table in the code. Every time the priority level 1 interrupt is triggered, the CPU runs through this table. Each byte will have a value that represents the time the corresponding LED should be ON. If the value is 0, it means that this LED is not turned ON. Every module will update this table when it is necessary, to reflect the current status of the module.

2.9 Embedded Debugger Implementation

The functional safety demonstrator has a mini Embedded Debugger (mEDBG) that can be used to program and debug ATtiny3217 using Unified Program and Debug Interface (UPDI). The mEDBG also includes a Virtual Com port interface over UART. The device can stream data that can be viewed using a serial port or the data visualizer in Atmel Studio. By shorting test points TP29 and TP30 on FSFEB, it is possible to use the mEDBG to program the board controller.

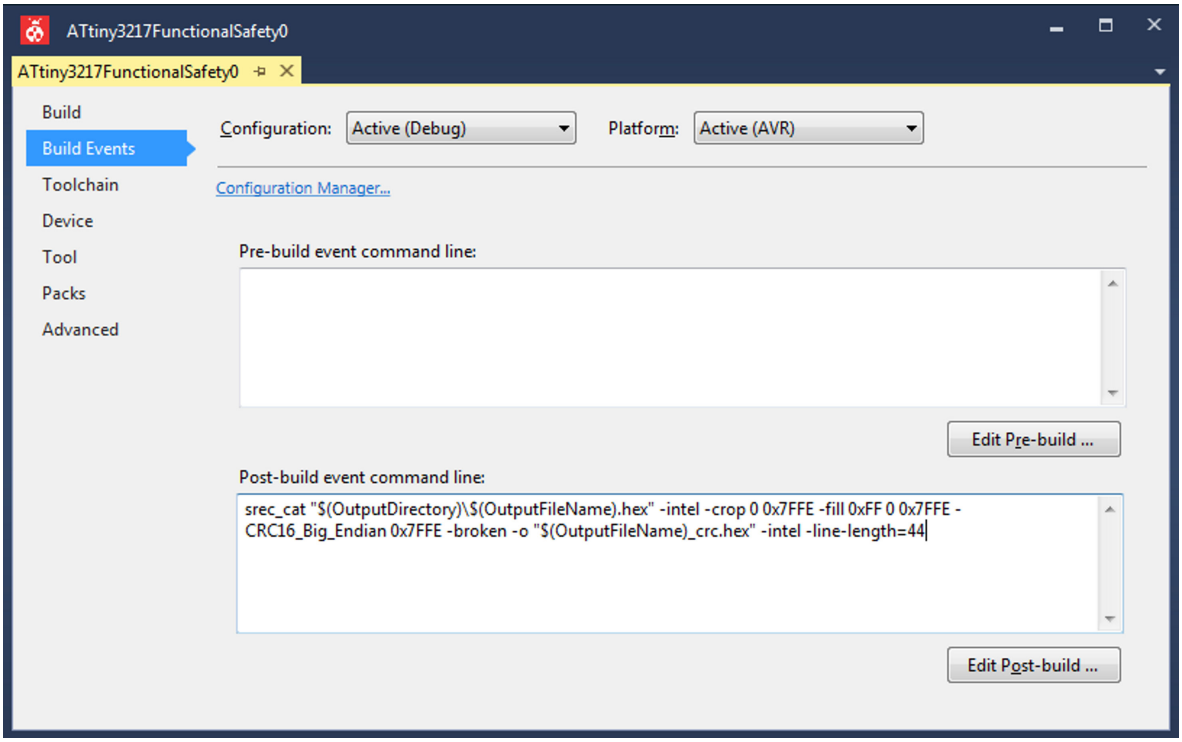
2.10 CRC Checksum Setup

The CRCSCAN peripheral expects a two byte CRC checksum to be located at the end of the area selected for scanning. This means that a CRC checksum needs to be calculated and added to the end of the flash image before the application is programmed to the device.

The tool srec_input can be used to generate the checksum and add it to the hex file. This is done in Atmel Studio, by adding a post-build command for the project. The end result will be a hex file which contains the application code and the CRC checksum. The hex file will be padded with *0xFF* to match the size of the device. To add the post-build command, do the following steps:

1. Open the project properties by clicking *Project → <project_name> Properties*.
2. Add a post-build command to the project under *Build Events → Post-build event command line*, as shown in the figure below.

Figure 2-16. Post-Build Command in Atmel Studio



3. Rebuild the project by clicking *Build → Rebuild <project_name>*.

The command to add is:

srec_cat "$(OutputDirectory)\$(OutputFileName).hex" -intel -crop 0 0x7FFE -fill 0xFF 0 0x7FFE -
CRC16_Big_Endian 0x7FFE -broken -o "$(OutputFileName)_crc.hex" -intel -line-length=44

Note: Take care to program the hex file with the appended CRC checksum to the device. Use the *device programming* dialog to program the device. The name of the file with the CRC checksum will end with *_crc.hex*. If the name of the project is left unchanged during creation, the filename will be *ATtiny3217FunctionalSafety0_crc.hex*.

3. Get Source Code from Atmel | START

The example code is available through Atmel | START, which is a web-based tool that enables configuration of application code through a Graphical User Interface (GUI). The code can be downloaded for both Atmel Studio and IAR Embedded Workbench® via the direct example code-link(s) below or the *BROWSE EXAMPLES* button on the Atmel | START front page.

Atmel | START web page: http://microchip.com/start

Example Code

- ATtiny3217 Functional Safety:
  - http://www.microchip.com/start/#example/Atmel:attiny3217_functional_safety:1.0.0::Application:ATtiny3217_Functional_Safety:

Press *User guide* in Atmel | START for details and information about example projects. The *User guide* button can be found in the example browser, and by clicking the project name in the dashboard view within the Atmel | START project configurator.

Atmel Studio

Download the code as an .atzip file for Atmel Studio from the example browser in Atmel | START, by clicking *DOWNLOAD SELECTED EXAMPLE*. To download the file from within Atmel | START, click *EXPORT PROJECT* followed by *DOWNLOAD PACK*.

Double-click the downloaded .atzip file and the project will be imported to Atmel Studio 7.0.

IAR Embedded Workbench

For information on how to import the project in IAR Embedded Workbench, open the Atmel | START user guide, select *Using Atmel Start Output in External Tools*, and *IAR Embedded Workbench*. A link to the Atmel | START user guide can be found by clicking *About* from the Atmel | START front page or *Help And Support* within the project configurator, both located in the upper right corner of the page.

4. Revision History

Doc. Rev.	Date	Comments
A	06/2018	Initial document release.

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: 978-1-5224-3188-6

Quality Management System Certified by DNV

ISO/TS 16949

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

Worldwide Sales and Service

AMERICAS	ASIA/PACIFIC	ASIA/PACIFIC	EUROPE
Corporate Office 2355 West Chandler Blvd. Chandler, AZ 85224-6199 Tel: 480-792-7200 Fax: 480-792-7277 Technical Support: http://www.microchip.com/support Web Address: www.microchip.com **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	**Australia - Sydney** Tel: 61-2-9868-6733 **China - Beijing** Tel: 86-10-8569-7000 **China - Chengdu** Tel: 86-28-8665-5511 **China - Chongqing** Tel: 86-23-8980-9588 **China - Dongguan** Tel: 86-769-8702-9880 **China - Guangzhou** Tel: 86-20-8755-8029 **China - Hangzhou** Tel: 86-571-8792-8115 **China - Hong Kong SAR** Tel: 852-2943-5100 **China - Nanjing** Tel: 86-25-8473-2460 **China - Qingdao** Tel: 86-532-8502-7355 **China - Shanghai** Tel: 86-21-3326-8000 **China - Shenyang** Tel: 86-24-2334-2829 **China - Shenzhen** Tel: 86-755-8864-2200 **China - Suzhou** Tel: 86-186-6233-1526 **China - Wuhan** Tel: 86-27-5980-5300 **China - Xian** Tel: 86-29-8833-7252 **China - Xiamen** Tel: 86-592-2388138 **China - Zhuhai** Tel: 86-756-3210040	**India - Bangalore** Tel: 91-80-3090-4444 **India - New Delhi** Tel: 91-11-4160-8631 **India - Pune** Tel: 91-20-4121-0141 **Japan - Osaka** Tel: 81-6-6152-7160 **Japan - Tokyo** Tel: 81-3-6880- 3770 **Korea - Daegu** Tel: 82-53-744-4301 **Korea - Seoul** Tel: 82-2-554-7200 **Malaysia - Kuala Lumpur** Tel: 60-3-7651-7906 **Malaysia - Penang** Tel: 60-4-227-8870 **Philippines - Manila** Tel: 63-2-634-9065 **Singapore** Tel: 65-6334-8870 **Taiwan - Hsin Chu** Tel: 886-3-577-8366 **Taiwan - Kaohsiung** Tel: 886-7-213-7830 **Taiwan - Taipei** Tel: 886-2-2508-8600 **Thailand - Bangkok** Tel: 66-2-694-1351 **Vietnam - Ho Chi Minh** Tel: 84-28-5448-2100	**Austria - Wels** Tel: 43-7242-2244-39 Fax: 43-7242-2244-393 **Denmark - Copenhagen** Tel: 45-4450-2828 Fax: 45-4485-2829 **Finland - Espoo** Tel: 358-9-4520-820 **France - Paris** Tel: 33-1-69-53-63-20 Fax: 33-1-69-30-90-79 **Germany - Garching** Tel: 49-8931-9700 **Germany - Haan** Tel: 49-2129-3766400 **Germany - Heilbronn** Tel: 49-7131-67-3636 **Germany - Karlsruhe** Tel: 49-721-625370 **Germany - Munich** Tel: 49-89-627-144-0 Fax: 49-89-627-144-44 **Germany - Rosenheim** Tel: 49-8031-354-560 **Israel - Ra'anana** Tel: 972-9-744-7705 **Italy - Milan** Tel: 39-0331-742611 Fax: 39-0331-466781 **Italy - Padova** Tel: 39-049-7625286 **Netherlands - Drunen** Tel: 31-416-690399 Fax: 31-416-690340 **Norway - Trondheim** Tel: 47-7289-7561 **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