AN2945

SAM L10/L11 Family Configurable Custom Logic (CCL) Peripheral Implementation

Introduction

This Application Note describes various features of the Configurable Custom Logic (CCL) peripherals. This document comes along with an IR encoding example application which is applicable for the SAM L10/L11 family.

The SAM L10/L11 family has an embedded CCL peripheral, which contains programmable logic to implement logic gates, sequential logic, or a combination of both. This peripheral contains two programmable LookUp Tables (LUTs) that consist of the following three inputs:

- A truth table
- An optional synchronizer, filter
- An optional edge detector

Each LUT can generate an output from a user programmable logic expression, such as NAND, NOR, XOR, and personalized logic expression. Optional sequential logic can also be used with flip-flops and latches to enable complex waveform generation.

The CCL can serve as glue logic between the SAM L11 and external devices. The CCL can eliminate the need for external logic components and can also help the designer to overcome challenging real-time constraints. This is accomplished by combining core independent peripherals in clever ways to handle the most critical parts of the application independent of the CPU.

Table of Contents

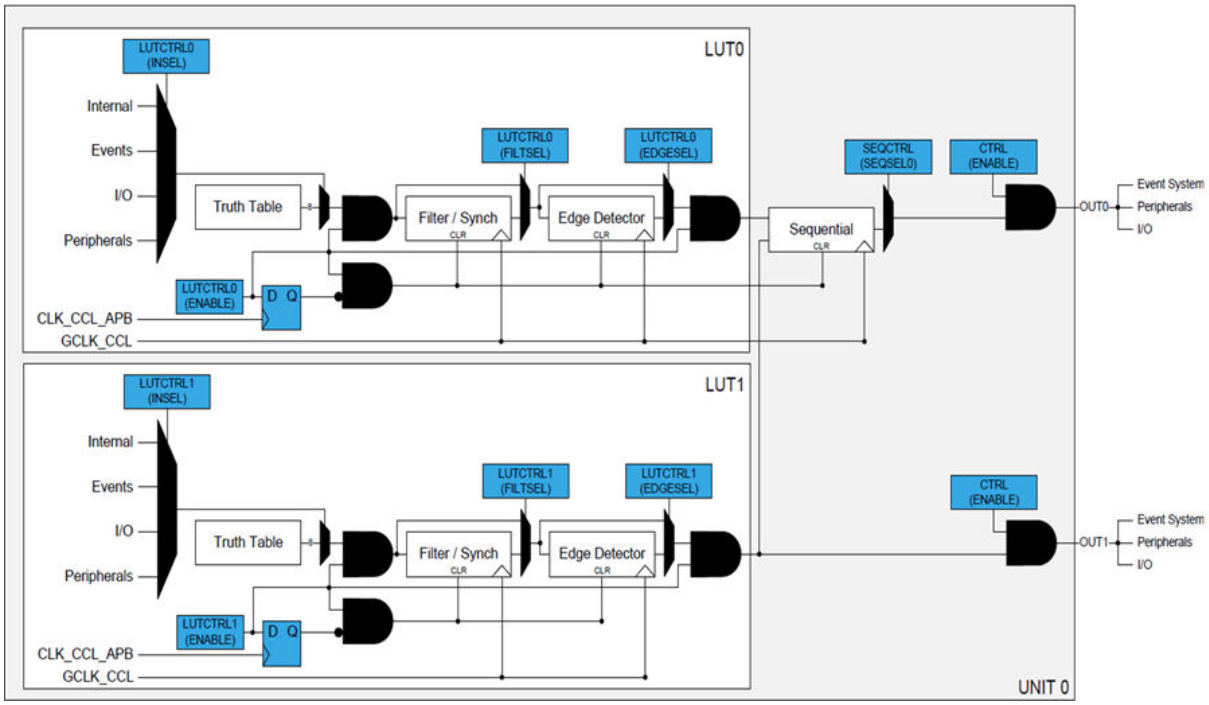
1. Peripheral Overview

The CCL peripheral features are as follows:

- Two LUTs with flexible input selection options:
  - I/Os
  - Events
  - Internal peripherals (AC, TC, SERCOM)
  - Feedbacks
  - Subsequent LUT output
  - Masked inputs
- Combinatorial logic functions: AND, NAND, OR, NOR, XOR, XNOR, NOT, or personalized logic function
- Sequential logic functions: Gated D flip-flop, JK flip-flop, gated D latch, RS latch
- Optional synchronizer, filter or edge detector available on each LUT output
- Output can be connected to the I/O pins or the Event System

The following figure illustrates the CCL block diagram.

Figure 1-1. CCL Block Diagram



Note: A unit contains two LUTs and the CCL peripheral contains all the units. In SAM L10/L11, CCL is composed of one Unit as there are two LUTs.

The following registers are used for configuring the CCL peripheral:

- The CCL Control Register (`CCL.CTRL.reg`) for generic peripheral settings
- The CCL LookUp Table register (`CCL.LUTCTRLx.reg`) to configure LUT and filter or edge detector
- CCL Sequential Control register (`CCL.SEQCTRL0.reg`) to select the sequential configuration

1.1 Control Register

The Control register enabless configuring the generic functions of the CCL peripheral. The Control register bit details are as follows:

- The CTRLx.ENABLE bit must be set to enable CCL.
- To enable the module to run in Standby mode, the CTRLx.RUNSTDBY bit must be set.
- When the CTRLx.SWRST bit is set, it proceeds to a CCL software reset.

1.2 Programmable LookUp Table (LUT)

LUT is made up the following three inputs and one output:

- One truth table
- A filter
- An edge detector

The LUTCTRLx register needs to be modified to configure each LUT component.

- The LUTCTRLx.INSELy bits must be modified to choose the type of LUT inputs.
- The LUTCTRLx.TRUTH[7:0] bits must be modified to choose the logic function applied at inputs.
- The LUTCTRLx.EDGESEL bit must be set to enable the edge detector.
- The LUTCTRLx.FILTSEL[1:0] bits must be modified to select the LUT output filter.
- The LUTCTRLx.ENABLE bit must be set to enable the LUT.

For additional information about the CCL.LUTCTRLx register, refer to the SAM L10/L11 Family Data Sheet (DS60001513).

1.3 Filter and Edge Detector

By default, the LUT output is a combinatorial function of the LUT inputs. This may cause some short glitches when the inputs change value. These glitches can be removed using filters, if required by the application. The filter selection bits in the LUT Control register (LUTCTRLx.FILTSEL) selects the synchronizer or digital filter options. When a filter is enabled, the output will be delayed by three to four GCLK_CCL clock cycles. All corresponding internal filter logic is cleared one APB clock cycle after a LUT is disabled.

The edge detector can be used to generate a pulse when detecting a rising edge on its input. To detect a falling edge, the truth table should be programmed to provide the opposite levels. The edge detector is enabled by writing a ‘1’ to the edge selection bit in the LUT Control register (LUTCTRLx.EDGESEL). In order to avoid unpredictable behavior, a valid filter option must be enabled.

For additional information about the filter and edge detector, refer to the CCL.LUTCTRLx register in the SAM L10/L11 Family Data Sheet (DS60001513).

1.4 Sequential Logic

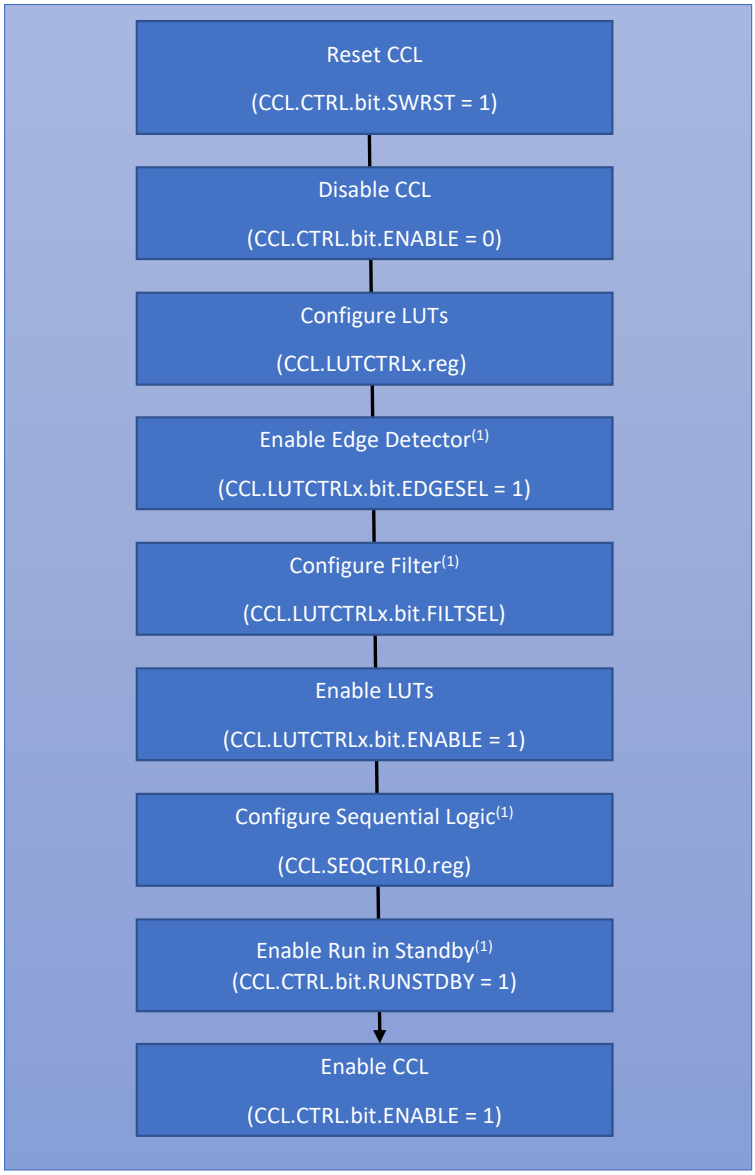
Each LUT pair can be connected to internal sequential logic, such as D flip flop, JK flip flop, gated D latch or RS latch, which can be selected by writing the corresponding Sequential Selection bits in the Sequential Control x register (SEQCTRLx.SEQSEL). Before using the sequential logic, the GCLK clock, and optionally each LUT filter, or edge detector must be enabled.

For additional information about the `CCL.SEQCTRLx` register, refer to the SAM L10/L11 Family Data Sheet (DS60001513).

All CCL registers can be configured in Atmel® START with a graphical overview. Refer to the section CCL Configuration for additional information.

The following flow chart illustrates how to configure the CCL peripheral:

Figure 1-2. CCL Configuration Flowchart



Note:

1. If required by the application.

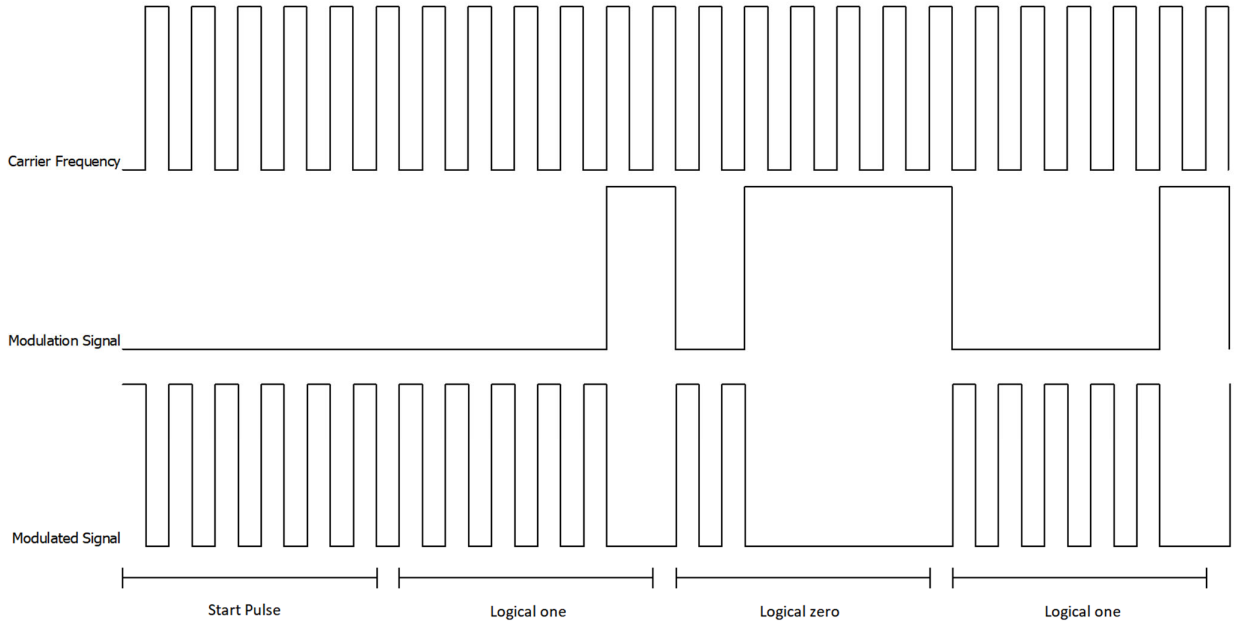
2. IR Encoding Example Overview

CCL can be used as a simple logic block for pin logic. In the following application example, a more advanced use case is presented. It also demonstrates how CCL can be used with TCs, combined with the Event System and DMA, to generate an IR encoded signal.

IR Encoding

There are many different standards for IR encoding. Most of them involve a fixed carrier frequency which will be on for a certain period and then off for a certain period. The enabling and disabling of the carrier frequency is determined by the duty cycle of a waveformed modulation signal. The duty cycle of the modulation signal will then determine if the transmitting value is a start pulse, a logical one, or a logical zero. For example, if the carrier frequency is signaled ¾ of the period, this indicates a logical one while ¼ indicates a logical zero as shown in the figure below:

Figure 2-1. IR Encoding



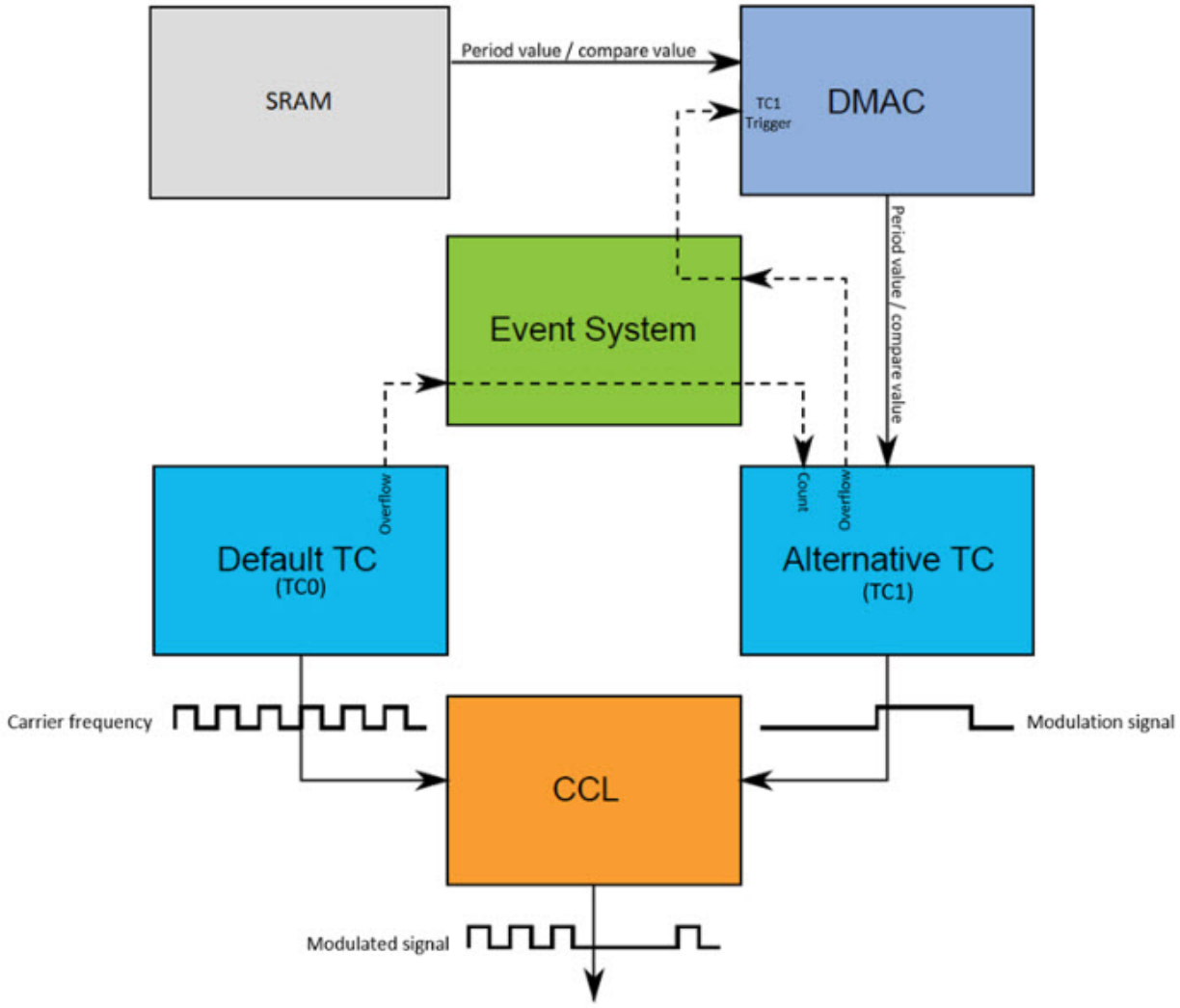
For the provided example application, no specific encoding is applied instead a general approach for where the period and duty cycle can be adjusted is presented.

Implementation

To generate a modulated IR signal, CCL is configured with two TC inputs while the last input is masked. The default TC (TC0) is used to generate the carrier frequency of the IR signal, while the alternative TC (TC1) is applied as the modulation signal. To ensure the modulation signal is synchronized with the carrier frequency, TC1 increments on overflow events generated by TC0. In this use case, the truth table is written 0x1 that corresponds to the NOR logic function. By adjusting the compare value for TC1, the duty cycle will change, resulting in longer or shorter intervals of transmitting the carrier frequency.

To update compare value and period, the DMAC is used with two channels transferring data from two arrays in the memory to the dedicated TC registers. Using the DMAC to update the compare value register and the period register allows the device to operate in Standby Sleep mode. The application flow is illustrated in the figure below:

Figure 2-2. Application Flow



2.1 Software and Hardware Requirements

The following software and hardware are required for the IR Encoding demonstration:

Software Requirements:

- Atmel Studio 7 (build 1931 or later)
- Atmel START
- SAM L10/L11 DFP version 1.0.81

Hardware Requirements:

- 1 x Microchip SAM L10 or SAM L11 Xplained Pro
- 1 x Micro USB cable (type-A or Micro-B)
- 1 x Oscilloscope or 1 x Logic Analyzer (optional)

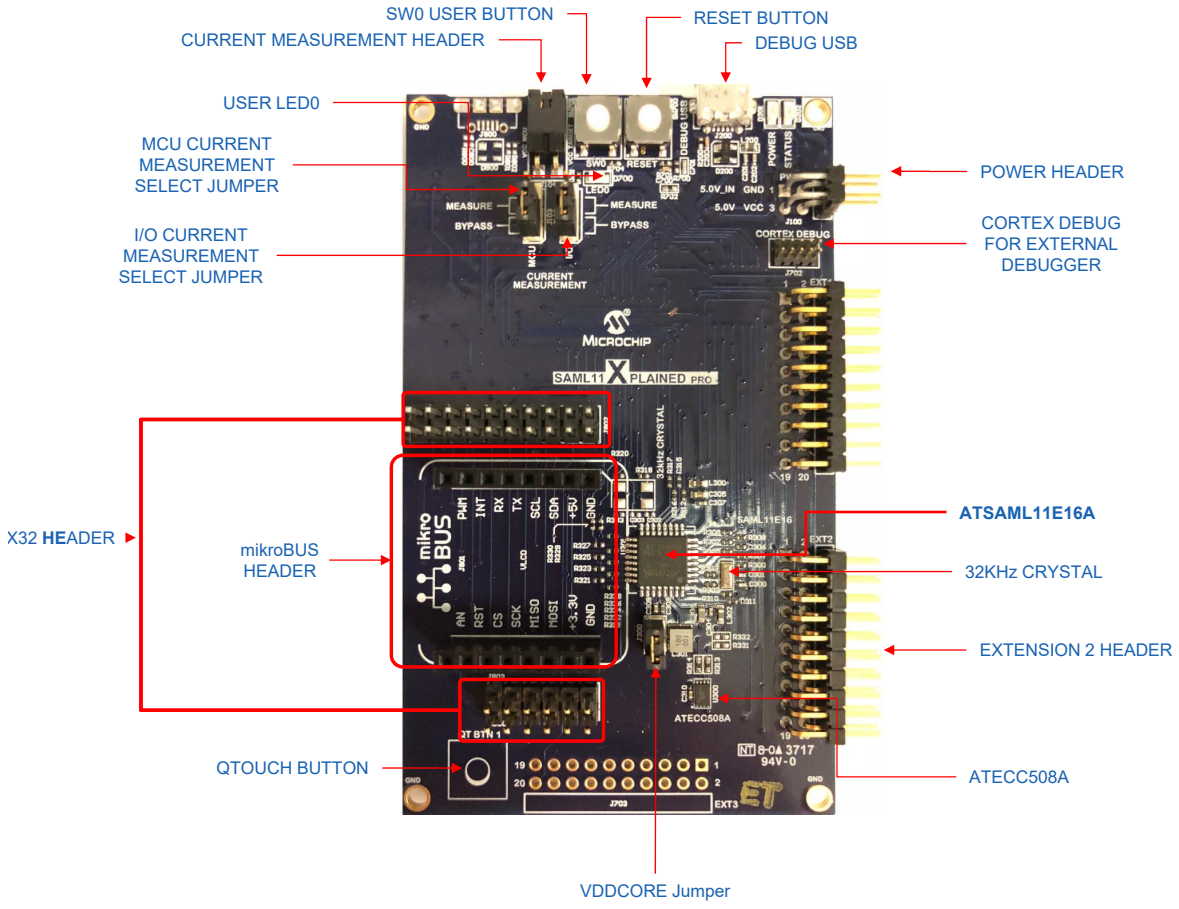
2.1.1 Hardware Requirements

2.1.1.1 SAM L10/L11 Xplained Pro Evaluation Kit

The Microchip SAM L10/L11 Xplained Pro evaluation kit is a hardware platform used to evaluate the ATSAML10E16A or ATSAMLL11E16A microcontroller. Supported by the Atmel Studio integrated development platform, the kit provides easy access to the features of the Microchip ATSAML10E16A or ATSAMLL11E16A, and explains how to integrate the device in a custom design.

The Xplained Pro MCU series evaluation kits include an onboard embedded debugger which overcome the need of external tools to program or debug the onboard microcontroller. The Xplained Pro extension kits offer additional peripherals to extend the features of the board, and ease the development of custom designs. The figure below illustrates the features of the SAM L10/L11 Xplained Pro board.

Figure 2-3. SAM L11 Xplained Pro Evaluation Kit



2.1.2 Software Requirements

2.1.2.1 Atmel® Studio 7 Integrated Development Platform

Figure 2-4. Atmel Studio 7

Atmel Studio 7 is the integrated development platform (IDP) for developing and debugging Atmel ARM® Cortex™-M processor-based and Atmel AVR® microcontroller applications.

The Atmel® Studio 7 IDP gives a seamless and easy-to-use environment to write, build, and debug your applications written in C/C++ or assembly code. Atmel Studio 7 supports all 8-bit and 32-bit AVR, the new SoC wireless family, SAM microcontrollers, and connects seamlessly to Atmel debuggers and development kits. Users can download Atmel Studio 7 from the following link: http://www.microchip.com/avr-support/atmel-studio-7.

2.1.2.2 Atmel® START

Figure 2-5. Atmel START

Atmel START is a web-based software configuration tool, for starting a new embedded development on Microchip SAM and AVR microcontrollers. Starting from either a new project or an example project, Atmel START enables you to select and configure a set of software components from SAM Advanced Software framework to tailor your embedded application in a usable and optimized manner. Atmel START supports code project generation for Atmel Studio 7, IAR Embedded Workbench® and Keil µVision®, or generic makefile generation.

2.2 Example Configuration

2.2.1 Hardware Setup

The IR encoding needs an oscilloscope or a logic analyzer to display the different signals generated. The following table shows which pins are used to plug-in the SAM L10/L11 Xplained Pro evaluation kit into one of these tools.

Table 2-1. SAM L10/L11 Pins Outputs

SAM L10/L11 Pins	Output signal
PA19	CCL Output (modulated signal)
PA22	TC0 Output (carrier frequency)
PA24	TC1 Output (modulation signal)

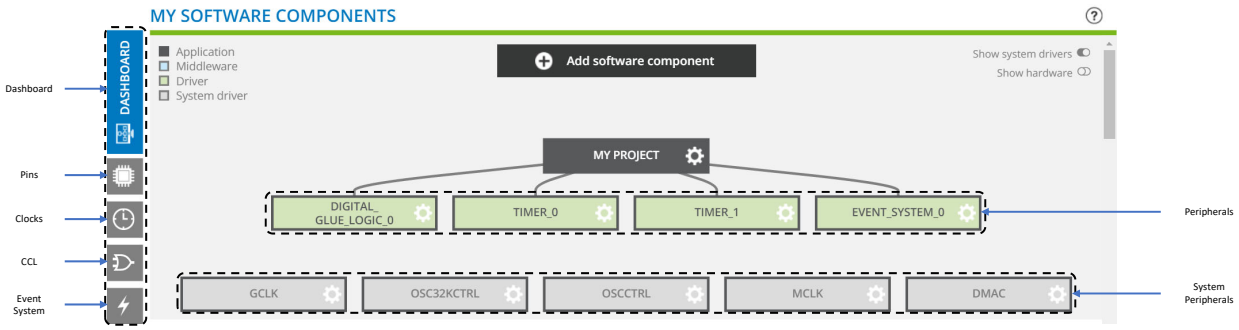
Refer to Generated Signals for signal details of these outputs.

2.2.2 Software Setup

For this example, many peripherals are used as shown in the following figure. The figure below shows the following peripherals:

- DIGITIAL_GLUE_LOGIC_0 (CCL)
- TIMER_0 (TC0)
- TIMER_1 (TC1)
- EVENT_SYSTEM_0 (EVSYS)

Figure 2-6. Atmel START Dashboard

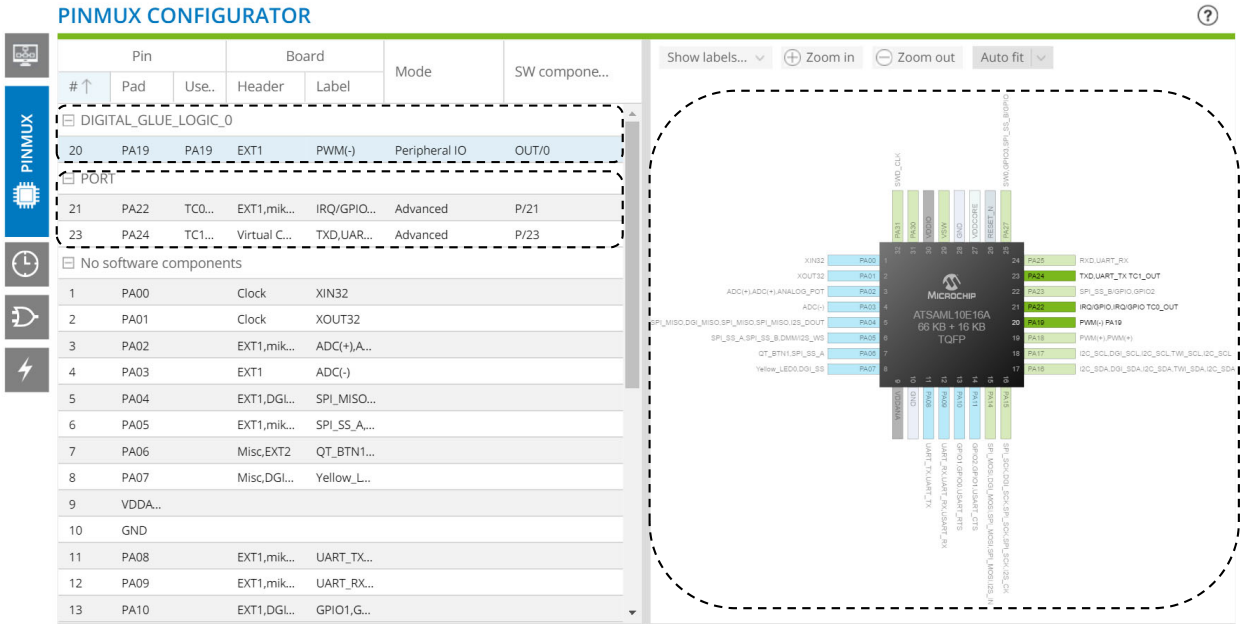


- Peripherals:
  - `DIGITAL_GLUE_LOGIC_0` (CCL) is configured to generate an IR encoded signal from `TIMER_0 (TC0)` and `TIMER_1 (TC1)`.
  - `TIMER_0` is configured to generate the carrier frequency for the IR Encoding.
  - `TIMER_1` is used to generate the modulation signal.
  - `EVENT_SYSTEM` is used to synchronize generated signals from TC0 an TC1.
- System Peripherals:
  - DMA moves the period and counter values from an array to TC1 period, and counter registers to generate the modulation signal.

Note: For SAM L11, TrustZone manager is required to run the application. Refer to the example provided with this document for more details.

2.2.2.1 Pins Configuration

Figure 2-7. Atmel START PINMUX



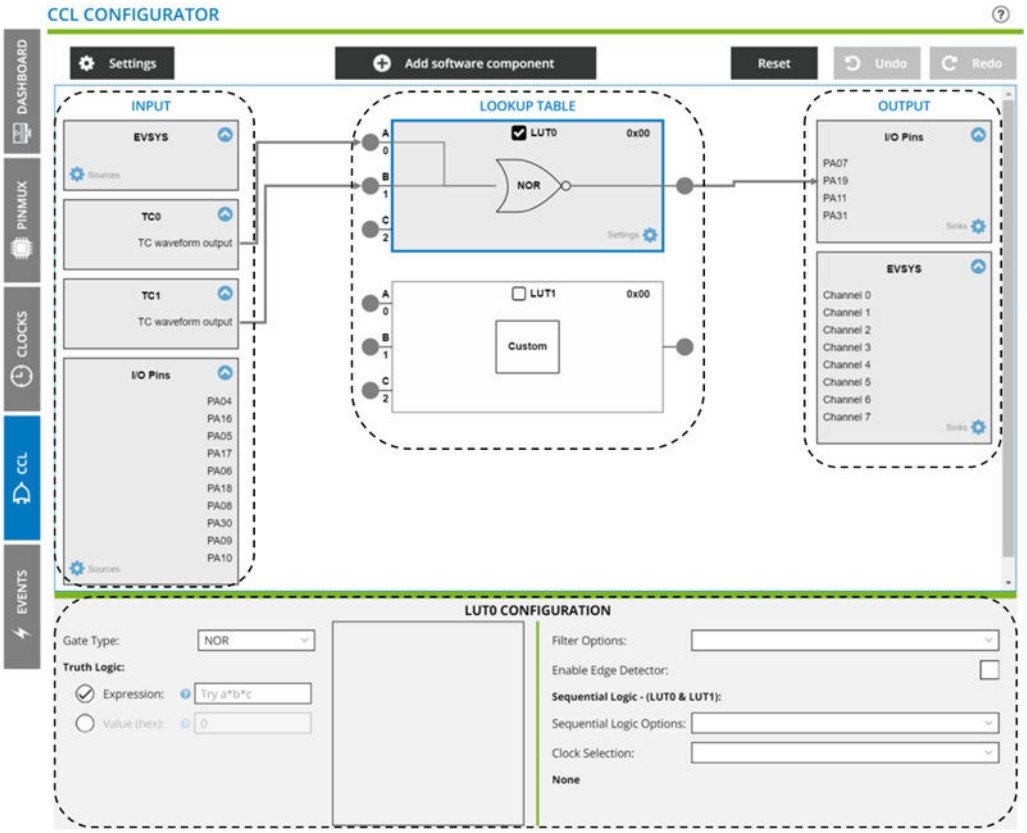
Configure the pins as shown below:

- PA19 is assigned to the CCL LUT0 output.
- PA22 is assigned to the TC0 output that generates the carrier frequency.
- PA24 is assigned to the TC1 output that illustrates the modulation signal.

2.2.2.2 CCL Configuration

Atmel START provides a graphical interface to configure the CCL peripheral. The following figure shows how CCL is configured for this use case.

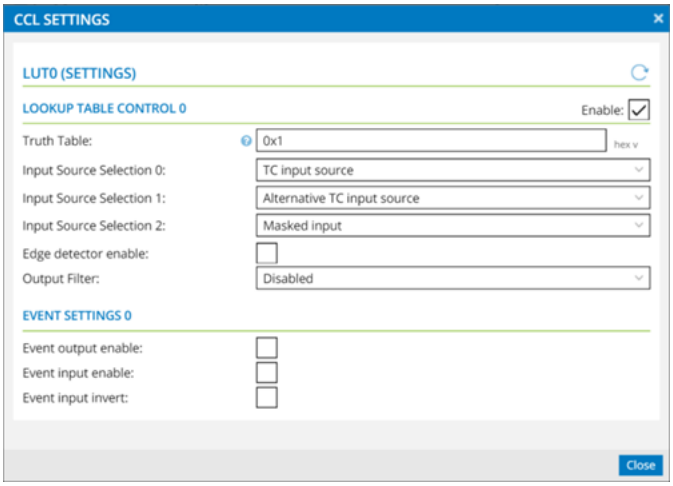
Figure 2-8. CCL Graphical Interface



The CCL graphical interface is divided into four parts: INPUT, LookUp Table, OUTPUT, and LUTx Configuration. Each of these parts provide specific information about the CCL configuration:

- INPUT enables selecting the following LUT input types:
  - TC0 is connected to LUT0 input 0
  - TC1 is connected to LUT0 input 1
- OUTPUT enables selecting the LUT output:
  - LUT0 output is connected to PA19
- LookUp Table enables the user to see inputs and logic functions that are used. In the LUT section, users can click the Settings tab to configure the LUTx. The figure below shows LUT0 setting details.

Figure 2-9. LUT0 Settings



  - LUT0 is enabled
  - Truth Table = 0x1 corresponds to a NOR logic gate
  - Input 0 is set as TC input source
  - Input 1 is configured as Alternative TC input source
  - Input 2 is masked
- LUTx Configuration enables to configure custom logic functions and sequential logic:
  - The gate type is a NOR logic gate
  - There is no filter, edge detector, or sequential logic used for this example

2.2.2.3 TCs Configuration

The following TCs are used as an input for LUT0, and these TCs are configured in Atmel START.

- TC0 is configured to generate the carrier frequency.
- TC1 is used as the modulation signal.

The Table below provides TC0 and TC1 configuration details.

Table 2-2. TC0 and TC1 Configurations

TC0 Configuration	TC1 Configuration
• 8-bit Counter Mode • Clock source: GCLK1 (runs at 142.857 kHz) • Default Capture mode • Prescaler: 1 • On demand clock • Run in Standby • Prescaler and Counter Synchronization: GCLK • Operating mode: 0x1 • Overflow Event Output enable • Normal PWM • Period value: 0x1 • Counter value: 0x0 • Compare/Capture Value 0: 0x1	• 8-bit Counter mode • Clock source: GCLK1 (runs at 142.857 kHz) • Default Capture mode • Prescaler: 1 • On demand clock • Run in Standby • Prescaler and Counter Synchronization: RESYNC • Operating mode: 0x1 • Overflow Event Output enable • TC Envent Input enable • Event action: COUNT • Normal PWM • Period value: 0x1 • Counter value: 0x0 • Compare/Capture Value 0: 0x1

2.2.2.4 DMAC Configuration

To use the DMA for transferring data, it is necessary to configure a DMA resource and transfer descriptor. The DMA resource contains the peripheral trigger, the trigger action, and channel configurations. The descriptor contains information regarding the specific data transfer, such as source and destination addresses, data size, next descriptor address, transfer counter and so on.

In this example, two different channels are used to write to the TC1 registers. For this purpose, one resource and one descriptor are needed for each channel as shown in the table below. The resources are configured in Atmel START, and the descriptors are configured with specific ASF4 functions from Atmel Studio, as shown in figure below.

DMA Channel 0 and 1 Resources Configuration
• Channel is enabled and Run in Standby • Trigger required for each beat transfer • TC1 Overflow Trigger source • Channel priority 0 • No Event Input Action • Address Increment Step Size: 1 • Step size is applied to the source address • Source Address Increment enabled • 8-bit bus transfer • No Block Action • No event generation

Figure 2-10. DMAC Channel 0 Descriptor Configuration

```
/*** Configure DMA Channel 0 ***/
void configure_dma_channel_0 (void)
{
    /* Set DMA Channel 0 source address */
    _dma_set_source_address(0, (void *)period_buffer);

    /* Set DMA Channel 0 destination address */
    _dma_set_destination_address(0, (void *)&TC1->COUNT8.PER.reg);

    /* Set DMA Channel 0 block length */
    _dma_set_data_amount(0, (uint8_t)BUF_LENGTH);

    /* Set DMA Channel 0 next descriptor */
    _dma_set_next_descriptor(0,0);
```

- `_dma_set_source_address()` sets the `period_buffer` table first value address as the source address for DMA Channel 0.
- `_dma_set_destination_address()` sets the TC1 8-bit Counter mode Period register as the destination address for the period value from the DMA source address.
- `_dma_set_data_amount()` indicates the `period_buffer` table length.
- `_dma_set_next_descriptor()` indicates the next descriptor that will be linked (channel 1 will loop with itself).

Figure 2-11. DMA Channel 1 Descriptor Configuration

```
/*** Configure DMA Channel 1 ***/
void configure_dma_channel_1 (void)
{
    /* Set DMA Channel 1 source address */
    _dma_set_source_address(1, (void *)value_buffer);

    /* Set DMA Channel 1 destination address */
    _dma_set_destination_address(1, (void *)&TC1->COUNT8.CC[0].reg);

    /* Set DMA Channel 1 block length */
    _dma_set_data_amount(1, (uint8_t)BUF_LENGTH);

    /* Set DMA Channel 1 next descriptor */
    _dma_set_next_descriptor(1,1);
}
```

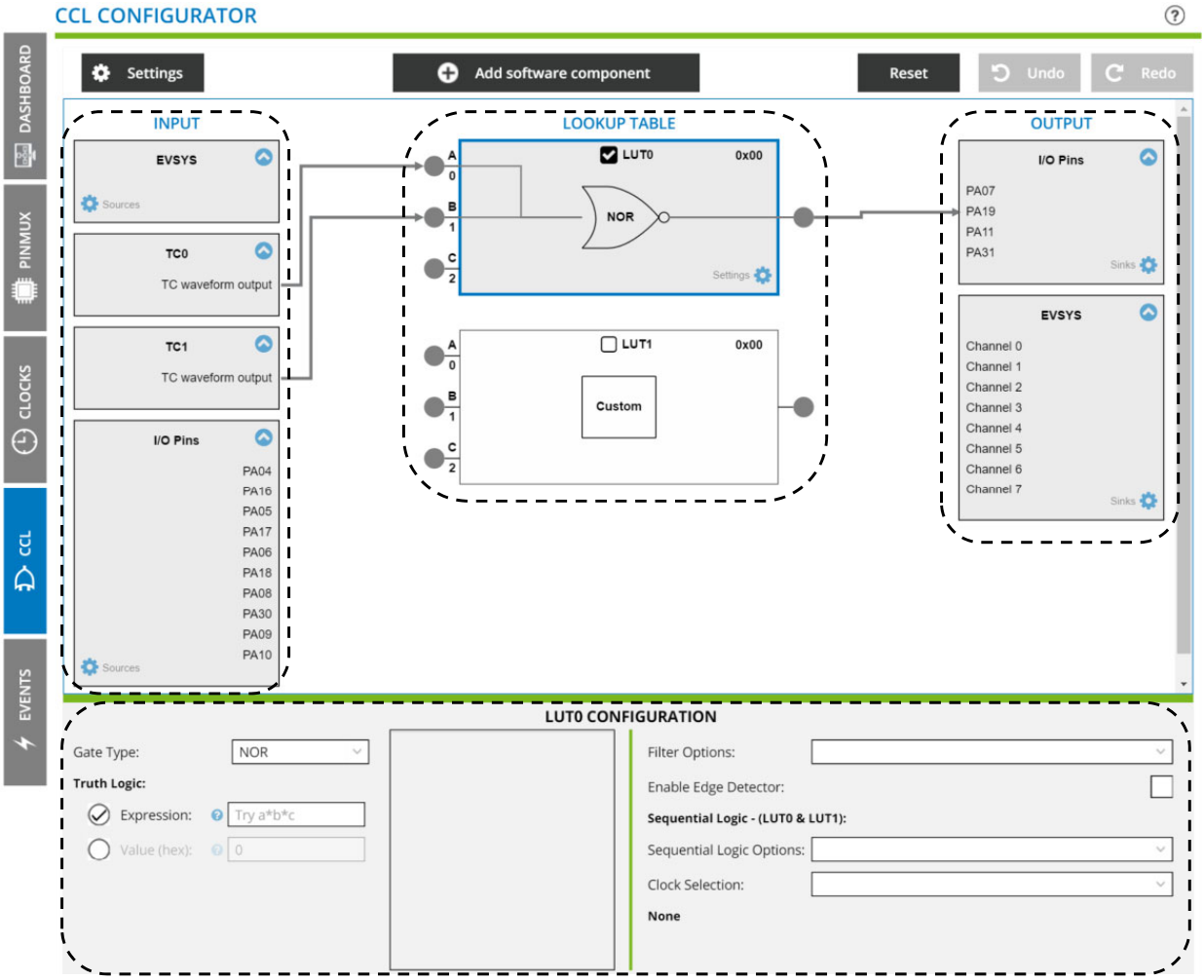
- `_dma_set_source_address()` sets the `value_buffer` table first value address as the source address for DMA Channel 0.
- `_dma_set_destination_address()` sets the TC1 8-bit Counter mode Compare/Counter register as the destination address for the period value from the DMA source address.

For additional information about DMA configuration, refer to the application example provided with this document.

2.2.2.5 Clock Configuration

The figure below illustrates the clock configuration.

Figure 2-12. Clock Configuration

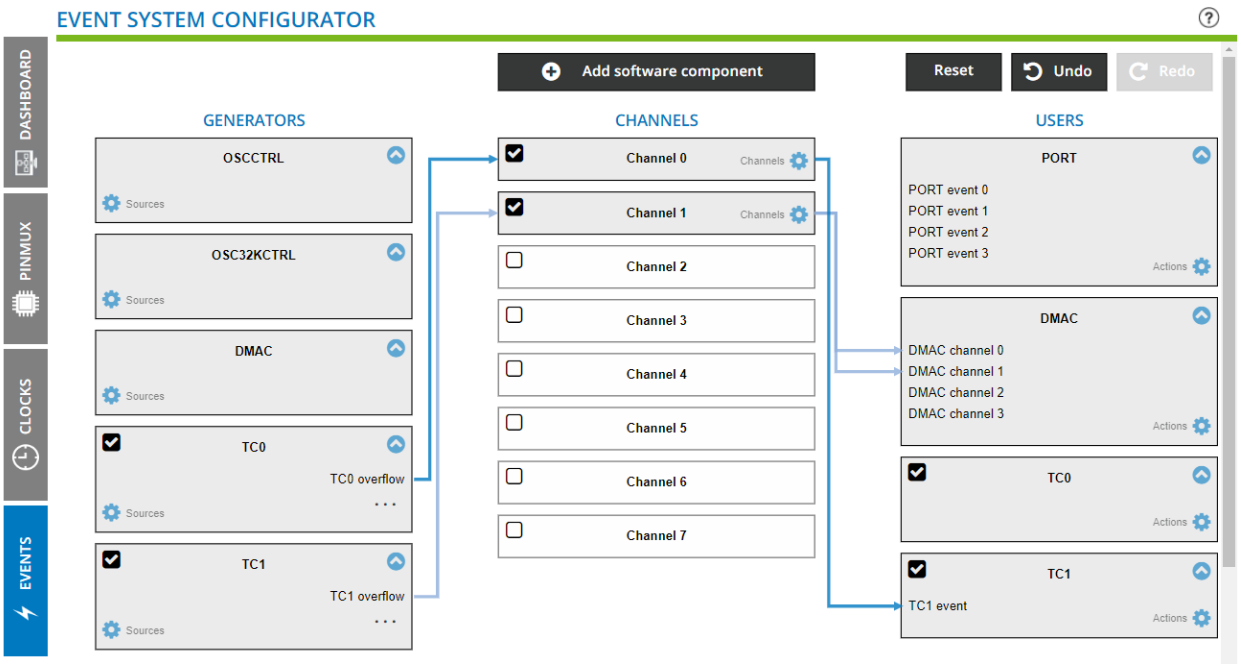


- OSC16M (16 MHz oscillator) is configured to run at 4 MHz and feeds the Generic Clock Generator 0 (GCLK0) and Generic Clock Generator 1 (GCLK1). GCLK0 runs at 4 MHz and GCLK1 runs at 142.857 kHz.
- The Generic Clock Controller (GCLK) is used to route oscillators to the peripherals. GCLK0 is used to clock the CPU and GCLK1 is used to clock the CCL, TC0, TC1, and Event System.

2.2.2.6 Event System (EVSYS) Configuration

EVSYS can be configured on Atmel START. In this example, event generation on overflow is enabled for both TCs. TC1 is also set to count on incoming events and the DMA is set to trigger a transfer on incoming events. Atmel START EVSYS is shown in the figure below.

Figure 2-13. Atmel START EVSYS



- Generators are TC0 overflow and TC1 overflow
- Channel 0 is enabled:
  - INPUT: TC0 overflow
  - OUTPUT: TC1 count event
  - Path selection: Asynchronous
- Channel 1 is enabled:
  - INPUT: TC1 overflow
  - OUTPUT: DMAC channel 0 and channel 1
  - Path selection: Resynchronized

2.2.2.7 Generated Files Configuration

After configuring the peripherals users can open the example project generated from Atmel START. The figure below shows the different functions used in the main routine to run the application:

Figure 2-14. Main Routine Function

```
int main(void)
{
    /* Initializes MCU, drivers and middleware */
    atmel_start_init();

    /* Configure DMA Channels 0 and 1 */
    configure_dmac();

    /* Enable CCL module */
    custom_logic_enable();

    /* Set BUCK as voltage regulator */
    SUPC->VREG.bit.SEL = SUPC_VREG_SEL_BUCK_Val;

    /* Activate Low Power Features */
    SUPC->VREG.bit.RUNSTDBY = 0x1;
    SUPC->VREG.bit.STDBYPL0 = 0x1;

    /* Disable the BOD33 */
    SUPC->BOD33.reg &= ~SUPC_BOD33_ENABLE;

    /* Enter in STANDBY mode */
    enter_standby_mode();

    while (1) {
    }
}
```

- `Atmel_start_init()` function initializes the MCU but also the CCL, TCs, DMA, Event System, clocks and pins according to Atmel START settings.
- `Configure_dmac()` allows the user to set the DMA descriptor values.
- To enable the CCL module in the main routine using Atmel START, the user must use the `custom_logic_enable()` function that will set the `CCLx.CTRL.ENABLE` bit to 1 as shown in the figure below:

Figure 2-15. CCL Enable Code

```
static inline void hri_ccl_set_CTRL_ENABLE_bit(const void *const hw)
{
    CCL_CRITICAL_SECTION_ENTER();
    ((Ccl *)hw)->CTRL.reg |= CCL_CTRL_ENABLE_Msk;
    CCL_CRITICAL_SECTION_LEAVE();
}
```

Note: The `hri_ccl_set_CTRL_ENABLE_bit ()` function is used by the `custom_logic_enable()` function to enable the CCL.

- BUCK is set as voltage regulator to reduce power consumptions.
- Voltage regulator runs in low power mode at PL0 during Standby Sleep mode.
- The Brown-out-Detector (BOD) is disabled to reduce power consumption.
- The `Enter_standby_mode()` function puts the device in Standby Sleep mode, and then continues generating the encoded signal, while the core is in Sleep mode.

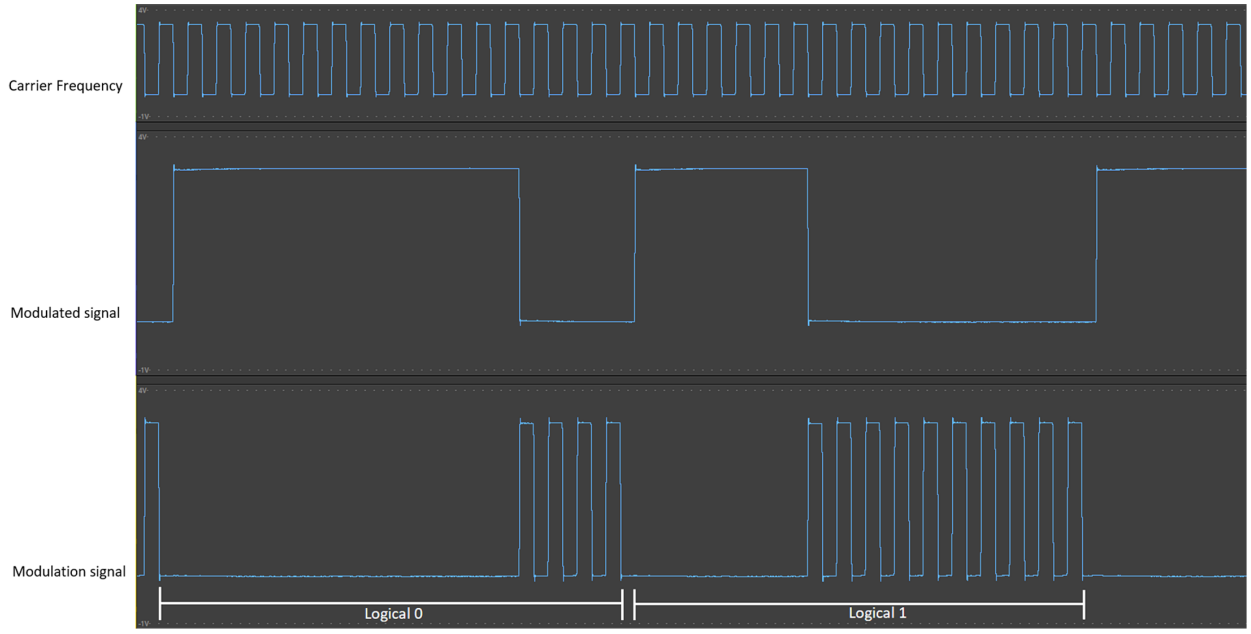
Note: Because the CCL runs independently from the CPU, it must be enabled to work. This explains the infinite loop.

2.3 Example Results

2.3.1 Generated Signals

When application is running, the following signals can be seen on the pins, PA19, PA22, and PA24, by connecting them to a signal analyzer:

Figure 2-16. Generated Signals



2.3.2 Power Consumption

When application is running, the dynamic current consumption of the whole application can be measured with the Data Visualizer tool from Atmel Studio. To reduce the current consumption, SAM L10/L11 is placed in Standby mode. It enables SAM L10/L11 to run a peripheral without waking up the CPU, and gives the same results with less power consumption.

The following figures show power consumption in Active mode and Standby mode:

Figure 2-17. Power Consumption in Active Mode

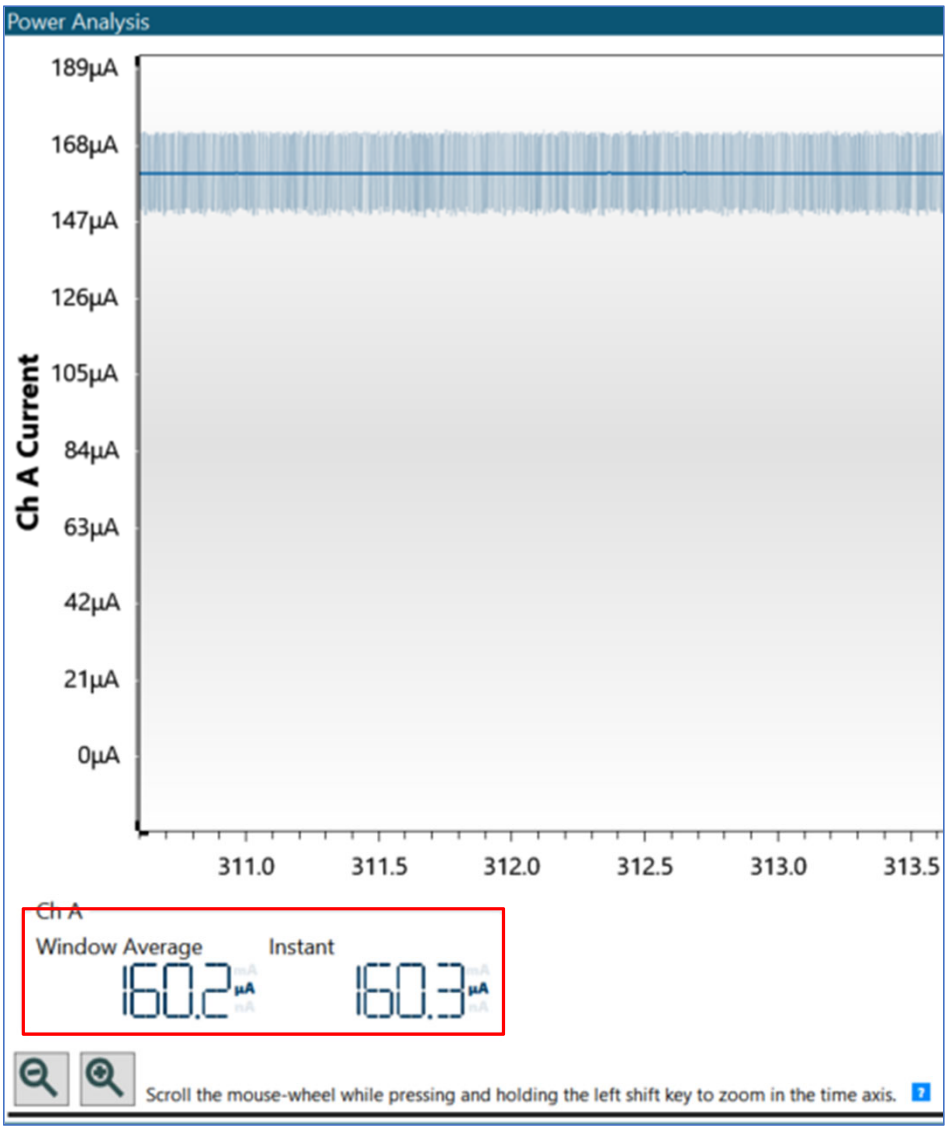
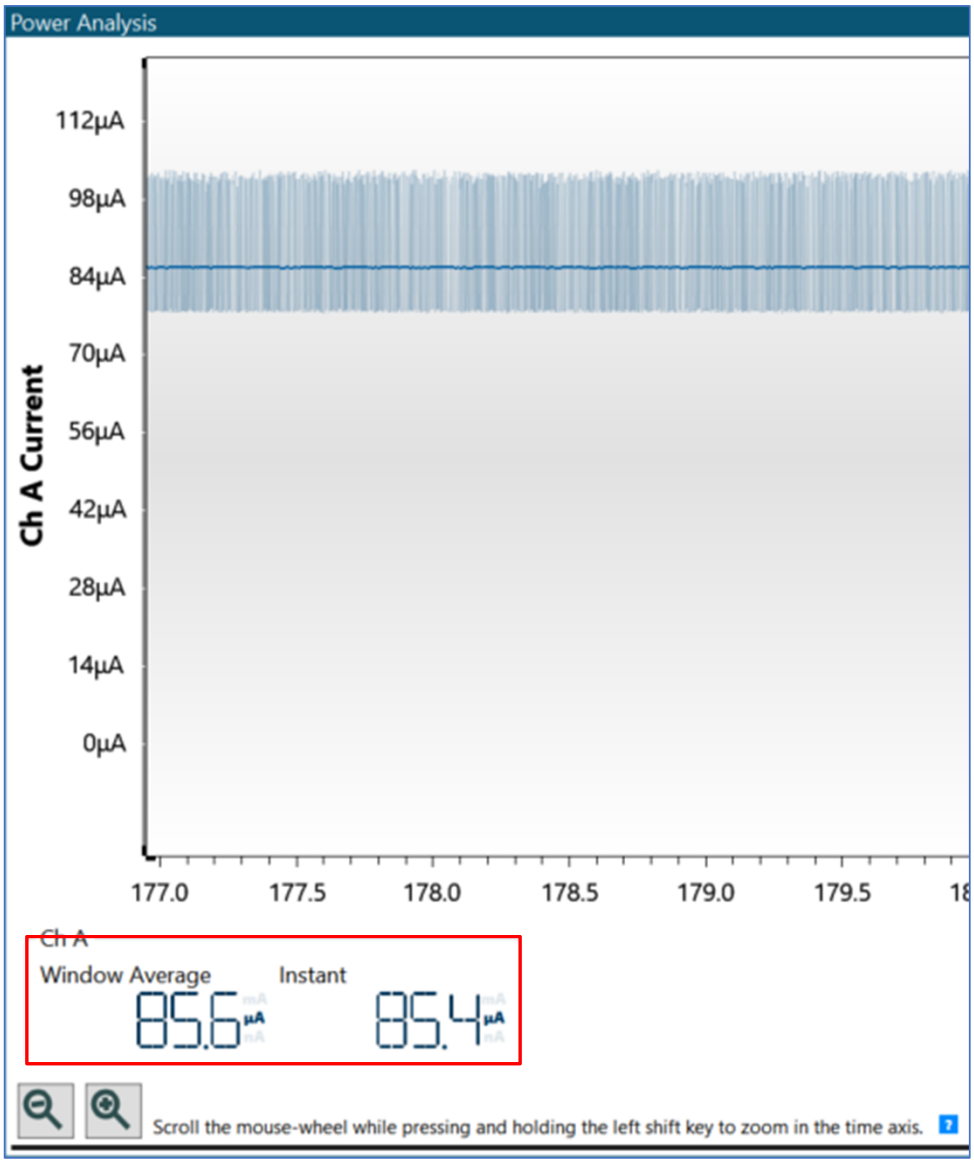


Figure 2-18. Power Consumption in Standby Mode



When entering Standby mode, it is possible to reduce power consumptions by half as shown in the images above.

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: 978-1-5224-4189-2

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: 47-72884388 **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