# Porting Considerations from C8051F30x and C8051F330/1/2/3/4/5 to C8051F85x/86x

## 1. Introduction

This application note describes how to migrate code from C8051F30x and C8051F330/1/2/3/4/5 devices to the C8051F85x/86x family. While there are some feature differences between the C8051F30x, C8051F330/1/2/3/4/5, and C8051F85x/86x devices, many applications can take advantage of the low-cost C8051F85x/86x family.

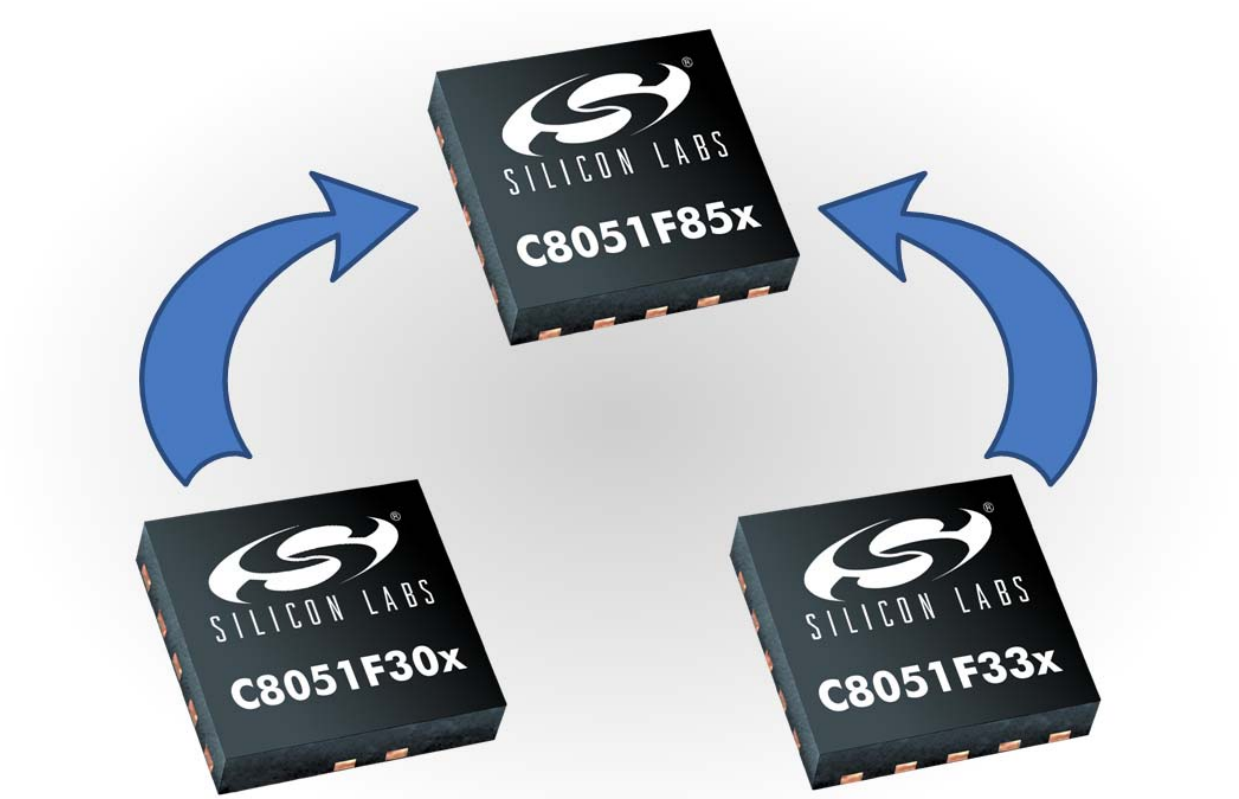



**Figure 1. Migrating Code from C8051F30x or C8051F330/1/2/3/4/5 to C8051F85x/86x Devices**

## 2. Relevant Documentation

Silicon Labs data sheets are available on the specific device landing page:

- **C8051F30x Data Sheet** — http://www.siliconlabs.com/products/mcu/smallmcu/Pages/C8051F30x.aspx and click on the **Documentation** tab.
- **C8051F33x Data Sheet —** http://www.siliconlabs.com/products/mcu/smallmcu/Pages/C8051F33x.aspx and click on the **Documentation** tab.
- **C8051F85x/86x Data Sheet —** http://www.siliconlabs.com/products/mcu/smallmcu/Pages/c8051f85x-86x.aspx and click on the **Documentation** tab.

# 3. New Features on the C8051F85x/86x Family

The C8051F85x/86x family of devices includes many features from the C8051F30x and C8051F330/1/2/3/4/5 devices, as well as enhancements and new features not available on the older parts. These features include:

- **ADC** — 8-bit, 10-bit, or 12-bit mode, burst mode, and automatic accumulation.
- **Voltage Reference** — Internal 1.65 V or 2.4 V reference.
- **Ports** — Programmable drive strength and port match.
- **Low Frequency Oscillator** — New module on the C8051F85x/86x devices.
- **SMBus** — Hardware acknowledge, adjustable start detection timing, and crossbar pin swapping.
- **Comparator** — An additional comparator module and auto-clear of the PCA CEXn outputs.
- **Programmable Counter Array** (PCA) — PWM enhancements (9-, 10-, 11-bit modes and center-aligned mode).
- **CRC** — New hardware block capable of performing general CRC calculations and automatically calculating the CRC of flash.
- **VDD Monitor** — Enabled by default after a power-on reset.
- **SPI** — Additional hardware block not available on the C8051F30x devices.
- **Device ID** — Identification for the family (C8051F85x/86x), device within the family (C8051F850), and hardware revision.
- **Watchdog Timer** — Separate module independent of the Programmable Counter Array (PCA).
- **Timer 2 and Timer 3** — Additional capture mode features.

The additional features for similar modules of the C8051F85x/86x family are generally backwards compatible. In other words, the new features are generally additions to the module as it exists on C8051F30x and C8051F330/1/2/3/4/5 devices.

## 3.1. ADC Enhancements

The C8051F85x/86x devices feature an ADC with configurable 8-bit (800 ksps), 10-bit (800 ksps), or 12-bit (200 ksps) modes. This ADC also has programmable gain (0.5x or 1x) and automatic accumulation of samples.

In addition, the C8051F85x/86x ADC has burst mode, where the ADC enters a low power state between conversions. For applications requiring lower sampling rates, using the C8051F85x/86x ADC in burst mode can result in considerable power savings. For example, the current consumption for 10-bit samples at 10 ksps is roughly 20 µA. The comparable power consumption for the C8051F30x and C8051F330/1/2/3/4/5 ADCs would be 400 µA, regardless of the sample rate.

## 3.2. Voltage Reference

The voltage reference on C8051F85x/86x devices is an internal reference with two programmable voltage levels: 1.65 V, and 2.4 V. The C8051F330/1/2/3/4/5 devices include a 2.4 V voltage reference only, and the C8051F30x devices do not include an internal reference.

## 3.3. Port I/O Enhancements

The Port I/O on the C8051F85x/86x devices have programmable drive modes. Each bank of ports (P0, P1, or P2) can be set to high drive or low drive mode. The low drive mode places the pins in a power-reducing configuration. High drive mode corresponds to the same drive capability as the C8051F30x and C8051F330/1/2/3/4/5 pins.

The C8051F85x/86x devices also have a port match feature that allows system events to be triggered by a logic value change on one or more port I/O pins. The PnMATCH registers specify the expected or normal logic values of the associated port pins. The PnMASK registers can be used to individually select which pins should be compared against the PnMATCH registers. A port mismatch event occurs if the logic levels of the port's input pins no longer match the PnMATCH registers, which can generate an interrupt, if enabled.

## 3.4. Low Frequency Oscillator

The C8051F85x/86x devices feature programmable low-frequency oscillator calibrated to a nominal frequency of 80 kHz. This oscillator can be divided by 1, 2, 4, or 8.

## 3.5. SMBus Enhancements

The C8051F85x/86x family includes three enhancements to the SMBus module: hardware acknowledge, adjustable start detection timing, and crossbar pin swapping. Other than these three enhancements, the SMBus module on the C805185x/86x family functions very similarly to the SMBus module on C8051F30x and C8051F330/1/2/3/4/5 devices and can operate in a backwards-compatible mode.

The hardware acknowledge feature enables automatic acknowledgement of a slave or general-call address while operating in slave mode. This feature is useful for systems with an SMBus or I2C master operating at higher frequencies (~300 kHz or above) that does not support clock stretching or in applications where there are many slaves on the bus besides this device.

In some systems where there is significant mis-match between the impedance or the capacitance on the SDA and SCL lines, it may be possible for SCL to fall after SDA during an address or data transfer. Such an event can cause a false start detection on the bus. These kind of events are not expected in a standard SMBus or I2C-compliant system, but the C8051F85x/86x devices have a programmable start detection hold time to prevent false start events in such systems.

The crossbar pin swapping feature of C8051F85x/86x devices enables the SDA and SCL pins to be swapped in hardware on the crossbar. In other words, the pins can be in either order: SDA first, SCL second or SCL first, SDA second.

## 3.6. Comparators

C8051F85x/86x devices include two comparators, whereas C8051F30x and C8051F330/1/2/3/4/5 devices only have one comparator. In addition, Comparator 0 on C8051F85x/86x devices have additional features to automatically clear the Programmable Counter Array (PCA) outputs. When this feature is enabled on a PCA channel, the corresponding CEX output is automatically set to 0 whenever the Comparator 0 output matches a programmable polarity.

## 3.7. Cyclic Redundancy Check Unit (CRC0)

C8051F85x/86x devices include a cyclic redundancy check unit (CRC0) that can perform a CRC using a 16-bit polynomial. CRC0 accepts a stream of 8-bit data written to the CRC0IN register, and CRC0 posts the 16-bit result to an internal register. The internal result register may be accessed indirectly using the CRCPNT bits and CRC0DAT register. CRC0 also has a bit-reverse register for quick data manipulation.

The CRC0 module is not available on C8051F30x and C8051F330/1/2/3/4/5 devices.

## 3.8. VDD Monitor

The VDD Monitor on C8051F85x/86x devices is enabled by default after a power-on reset. This enables the VDD Monitor to hold the device in reset for a longer period of time if VDD is not yet above the threshold as an added system security feature.

The VDD Monitor is not enabled by default after a power-on reset for C8051F30x and C8051F330/1/2/3/4/5 devices.

## 3.9. Serial Peripheral Interface (SPI)

The C8051F85x/86x devices include a Serial Peripheral Interface (SPI) module. This module is also present on the C8051F330/1/2/3/4/5 devices and is identical to the implementation on C8051F85x/86x devices. However, this module is not available on C8051F30x devices.

### 3.10. Device Identification

The C8051F85x/86x family has registers that identify the device family, derivative, and revision. These registers can be read by firmware at runtime to determine the specific device in the family or device revision that is executing code. This allows the same firmware image to run on MCUs with different memory sizes and peripherals and dynamically change functionality to suit the capabilities of that MCU.

The Device Identification registers are not available on C8051F30x and C8051F330/1/2/3/4/5 devices.

### 3.11. Watchdog Timer

On C8051F30x and C8051F330/1/2/3/4/5 devices, the Watchdog Timer is part of the Programmable Counter Array (PCA) module. Some registers of the PCA module may not be accessed by firmware while the Watchdog Timer is active. Firmware can clear the WDTE bit in the PCA0MD register to disable the Watchdog Timer.

On C8051F85x/86x devices, the Watchdog Timer is a separate module from the PCA, so the two can be used independently. The Watchdog Timer runs from the 80 kHz low-frequency oscillator and provides a timeout internal of up to 13.1 seconds. Firmware can disable the Watchdog Timer by writing the key sequence to the WDTCN register.

### 3.12. Timer 2 and Timer 3 Capture Modes

Timer 2 on C8051F85x/86x devices can capture against the external oscillator input frequency, while Timer 3 can capture against the low-frequency oscillator. On C8051F330/1/2/3/4/5 devices, Timer 2 and Timer 3 both capture against the external oscillator input frequency. C8051F30x devices do not support a timer capture feature.

## 4. Features Not Available on the C8051F85x/86x Family

Some features available on C8051F30x and C8051F330/1/2/3/4/5 devices are not available on C8051F85x/86x devices.

### 4.1. Specific ADC Features

The ADC on C8051F85x/86x devices is single-ended only and does not support differential measurements. In addition, the C8051F85x/86x ADC supports 0.5x and 1x gain, whereas the C8051F30x ADC supports 0.5x, 1x, 2x, and 4x gain. The C8051F330/1/2/3/4/5 devices do not have an internal programmable gain feature.

Applications using the C8051F30x or C8051F330/1/2/3/4/5 differential ADC feature would need to move to a single-ended configuration to use the C8051F85x/86x ADC. Firmware using the 2x or 4x gain settings with the 8-bit C8051F30x ADC achieve better or the same LSB size as the C8051F85x/86x 10-bit ADC (and much better LSB size in 12-bit mode) with 1x gain while increasing the measurable dynamic range.

### 4.2. External Oscillator

The C8051F85x/86x devices support external CMOS oscillators, but do not support C, RC, or crystal external oscillators. Both the C8051F30x and C8051F330/1/2/3/4/5 devices support these additional external oscillator configurations.

Applications requiring the external oscillator features of the C8051F30x and C8051F330/1/2/3/4/5 devices could swap to an external CMOS oscillator to migrate to the C8051F85x/86x family.

### 4.3. Current-Mode DAC (IDAC)

The C8051F330/1/2/3/4/5 devices include a current-mode DAC (IDAC). The IDAC module is not available on the C8051F85x/86x devices. As an alternative, applications that require a DAC can use an external module or a PCA channel in PWM mode with an external RC filter.

## 5. Feature Comparison Summary

Table 1 provides a summary for the primary differences among the three MCU families: C8051F30x, C8051F330/1/2/3/4/5, and C8051F85x/86x.

**Table 1. Feature Differences Between MCU Families**

| Feature | C8051F30x | C8051F330/1/2/3/4/5 | C8051F85x/86x |
|---|---|---|---|
| **Core** | | | |
| XRAM | — | 512 bytes | 256 bytes |
| **Analog — ADC** | | | |
| Resolution | 8 bits | 10 bits | 8, 10, or 12 bits |
| Maximum Throughput Rate | 500 ksps | 200 ksps | 800 ksps (8-bit)<br>800 ksps (10-bit)<br>200 ksps (12-bit) |
| Maximum SAR Conversion Clock | 6 MHz | 3 MHz | 12.5 MHz |
| Power Supply Current (VDD supplied to ADC0) | 400 µA (500 ksps) | 400 µA (200 ksps) | 845 µA (800 ksps)<br>370 µA (200 ksps, burst) |
| Burst Mode | — | — | ✓ |
| Automatic Accumulation | — | — | ✓ |
| Window Comparator | ✓ | ✓ | ✓ |
| Differential Inputs | ✓ | ✓ | — |
| Gain Settings | 0.5x, 1x, 2x, 4x | 1x | 0.5x, 1x |
| **Analog — Other** | | | |
| Voltage Reference | External only | External or 2.4 V internal that requires external capacitors | External or 1.65 V / 2.4 V internal that does not require external capacitors |
| Current-Mode DAC (IDAC) | — | 1 | — |
| Comparators | 1 | 1 | 2<br>Comparator 0 has additional PCA output auto-clear feature |
| VDD Monitor | ✓ | ✓ | enabled by default after a power-on reset |
| **Digital** | | | |
| Port Pins | 8 | 17 | 18 |
| Programmable Port Drive | — | — | ✓ |
| Port Match | — | — | ✓ |
| SMBus | ✓ | ✓ | additional hardware acknowledge, start detection timing, and crossbar pin swapping |
| Programmable Counter Array | ✓ | ✓ | additional 9-bit, 10-bit, and 11-bit PWM, center-aligned PWM |
| Timers | 3 | 4 | 4 |

**Table 1. Feature Differences Between MCU Families (Continued)**

| Feature | C8051F30x | C8051F330/1/2/3/4/5 | C8051F85x/86x |
|---|---|---|---|
| Timer Capture | — | External oscillator | External oscillator and low frequency oscillator |
| SPI | — | 1 | 1 |
| CRC (16-bit) | — | — | 1 |
| Device ID | — | — | ✓ |
| Watchdog Timer | part of PCA module | part of PCA module | independent module |
| Low Frequency Oscillator | — | — | ✓ |
| **Pinout and Packages** | | | |
| QFN-11 | ✓ | — | — |
| SOIC-14 | ✓ | — | — |
| QFN-20, 3x3 mm | — | — | ✓ |
| QFN-20, 4x4 mm | — | ✓ | — |
| SOIC-16 | — | — | ✓ |
| QSOP-24 | — | — | ✓ |

# 6. Migrating Firmware to the C8051F85x/86x Family

The C8051F30x, C8051F330/1/2/3/4/5, and C8051F85x/86x devices are different enough that code for one device must be recompiled with the correct header file to function properly on another device.

## 6.1. Special Function Register Differences

For features that exist on both the C8051F330/1/2/3/4/5 and C8051F85x/86x or C8051F30x and C8051F85x/86x, the registers are largely the same name and bits have roughly the same position. The registers with major differences are:

- **ADC0CN** — This register is now the ADC0CN0 register on C8051F85x/86x devices.
- **ADC0CF** — Bits for additional features are in this register for C8051F85x/86x devices.
- **CKCON** — Additional bits on C8051F85x/86x compared to C8051F30x to support the additional Timer 3.
- **CLKSEL/OSCICN** — Different clock configurations and frequency selections on all three device families (C8051F30x, C8051F330/1/2/3/4/5, and C8051F85x/86x). C8051F85x/86x devices do not have an OSCICN register, and the system clock divider settings are available in the CLKSEL register.
- **CPT0MD** — Additional bits for the Comparator 0 PCA auto-clear feature.
- **EIE1, EIP1** — These registers are slightly different to accommodate additional interrupt vectors on the C8051F85x/86x devices.
- **Multiplexer registers (CPT0MX, ADC0MX)** — These registers will be different only every family to support various packaging and pinout differences.
- **PCA0MD** — Contains Watchdog Timer controls for C8051F30x and C8051F330/1/2/3/4/5 devices. These controls are moved to the WDTCN register on C8051F85x/86x devices.
- **Ports (Pn, PnMDIN, PnMDOUT, etc.)** — This registers vary from family to family depending on the number of pins on the device.
- **REF0CN** — Modifications to support the 1.65 V / 2.4 V settings on the C8051F85x/86x devices.
- **TMR2CN** — Additional bits on C8051F85x/86x to support Timer 2 capture.
- **XBR0, XBR1, XBR2** — These registers have slightly shifted bits to accommodate different modules available on the crossbars.

## 6.2. Example Firmware Changes

This section lists examples of the code differences between the C8051F30x and C8051F330/1/2/3/4/5 to the C8051F85x/86x family. For an example of porting an ADC example, see "6.3. C8051F330/1/2/3/5 to C8051F85x/86x ADC Porting Example" on page 9.

### 6.2.1. Watchdog Timer

On the C8051F30x and C8051F330/1/2/3/4/5 devices, disable the watchdog timer with the following code:

```
PCA0MD &= ~0x40;
```

On the C8051F85x/86x devices, the same task is accomplished with the WDTCN register:

```
WDTCN = 0xDE;

WDTCN = 0xAD;
```

### 6.2.2. System Clock

To configure the system clock to use the 24.5 MHz High-Frequency Oscillator on the C8051F30x and C8051F330/1/2/3/4/5 devices:

```
OSCICN = 0x83;
```

On C8051F85x/86x devices:

```
CLKSEL = 0x00;
```

In addition, C8051F30x and C8051F330/1/2/3/4/5 must manually disable the 24.5 MHz oscillator if using a different oscillator as the system clock:

```
OSCICN = 0x03;
```

On C8051F85x/86x devices, the oscillator will automatically turn off whenever it's not needed.

### 6.2.3. Crossbar

To enable the crossbar and weak pull-ups on C8051F33x devices:

```
XBR1 = 0x40;
```

On C8051F30x and C8051F85x/86x devices, the equivalent code is:

```
XBR2 = 0x40;
```

### 6.2.4. Voltage Reference

The C8051F85x/86x devices have additional options to set the internal reference to 1.65 V or 2.4 V:

```
REF0CN = 0x18; // 1.65 V internal reference

REF0CN = 0x98; // 2.4 V internal reference
```

The 1.65 V option is not available on C8051F30x and C8051F330/1/2/3/4/5 devices.

### 6.2.5. Configuring ADC0

The code below sets ADC0 with the maximum conversion clock, default resolution, right justified, and 1x gain.

On C8051F30x, the code is:

```
ADC0CF = (((SYSCLK/6000000)-1)<<3) | 0x01;
```

On C8051F330/1/2/3/4/5 devices, the firmware to do this is:

```
ADC0CF = ((SYSCLK/3000000)-1)<<3;
```

On C8051F85x/86x devices, it's:

```
ADC0CF = (((SYSCLK/12250000)-1)<<3) | 0x01;
```

C8051F85x/86x devices additionally support burst mode and auto-accumulate, which are not available on C8051F30x and C8051F330/1/2/3/4/5 devices. To configure these options:

```
ADC0PWR = 0x44;    // Burst Mode Power Up Time

ADC0TK = 0x3A;     // Burst Mode Tracking Time

ADC0AC = 0x05;     // Accumulate enable and accumulate 64 conversions

ADC0CN0 |= 0x40;   // Enable burst mode
```

### 6.2.6. Comparator Multiplexer Selection

To set P0.1 as the negative input to Comparator 0 and set P0.0 as the positive input:

On C8051F30x and C8051F33x devices, the code is:

```
CPT0MX = 0x00;
```

On C8051F85x/86x devices, the equivalent code is:

```
CPT0MX = 0x10;
```

## 6.3. C8051F330/1/2/3/5 to C8051F85x/86x ADC Porting Example

This example ports an ADC example from the C8051F330/1/2/3/4/5 devices to the C8051F85x/86x devices. This example first disables the watchdog timer, configures the core to run at 24.5 MHz, and then configures the ADC to sample at a slow rate (10 ksps), averaging over 8 cycles and outputting the result on the UART at 115200 baud.

To port an ADC example from the C8051F330/1/2/3/4/5 devices to the C8051F85x/86x devices:

1. Prepare the necessary files:
   a. Make a folder for the new project. For the purposes of this example, we'll call the folder **Porting_Example**
   b. Copy the C8051F850_defs.h and si_toolchain.h header files into the **Porting_Example** folder. These files can be found in the ***C:\SiLabs\MCU\Examples\C8051F85x_86x\Header_Files*** directory by default after installing the 8-bit Silicon Labs IDE from www.siliconlabs.com/8bit-software.
   c. Copy the C8051F330/1/2/3/4/5 **F33x_ADC0_ExternalInput.c** example from ***C:\SiLabs\MCU\Examples\C8051F330_5\ADC*** into the **Porting_Example** folder. Rename the file, if desired. For this example, we will rename the file to **F85x_ADC0_porting_example.c**.
   d. Open the 8-bit IDE and select **Project→New Project...**, click **Cancel** on the new project window that appears, right-click on the **New_Project** folder, select **Add files to project New_Project,** and browse to **F85x_ADC0_porting_example.c**. Then, save the project by going to **Project→Save Project As...**.
2. Double-click on the **F85x_ADC0_porting_example.c** file to open it in the IDE editor window.
3. Change the header file include to the C8051F85x/86x header file.

| old | new | | |
|---|---|---|---|
| 115 | | – #include <c8051f330.h> | // SFR declarations |
| | 115 | + #include <si_toolchain.h> | |
| | 116 | + #include <C8051F850_defs.h> | // SFR declarations |

4. Disable the watchdog timer. The C8051F85x/86x family has a dedicated module and is disabled differently than the C8051F330/1/2/3/4/5.

| old | new | | |
|---|---|---|---|
| 150 | | – PCA0MD &= ~0x40; | // WDTE = 0 (clear watchdog timer |
| 151 | | – | // enable) |
| | 150 | + // Disable the watchdog timer -"DEAD" | |
| | 151 | + WDTCN = 0xDE; | |
| | 152 | + WDTCN = 0xAD; | |

5. Configure the core to run from the internal 24.5 MHz oscillator.

| old | new | | |
|---|---|---|---|
| 183 | | – OSCICN = 0x83; | // configure internal oscillator for |
| 184 | | – | // 24.5MHz |
| 185 | | – RSTSRC = 0x04; | // enable missing clock detector |
| | 185 | + CLKSEL = 0x00; | // Set system clock to 24.5 MHz |
| | 186 | + | |
| | 187 | + RSTSRC = 0x06; | // Enable missing clock detector |

6. Configure the ports and crossbar, setting P1.2 (ADC0.10) as an analog input.

| old | new | | |
|---|---|---|---|
| 204 | 206 | XBR0 = 0x01; | // Enable UART0 |
| 205 | | – XBR1 = 0x40; | // Enable crossbar and weak pull-ups |
| | 207 | + XBR2 = 0x40; | // Enable crossbar and weak pull-ups |
| 206 | 208 | P0MDOUT \|= 0x10; | // Set TX pin to push-pull |
| 207 | | – P1MDOUT \|= 0x08; | // enable LED as a push-pull output |
| 208 | | – P1MDIN &= ~0x10; | // set P1.4 as an analog input |
| | 209 | + P1MDOUT \|= 0x01; | // enable LED as a push-pull output |
| | 210 | + P1MDIN &= ~0x04; | // set P1.2 as an analog input |

7. Configure ADC0, setting the SAR clock to 12.25 MHz with the ADC in 10-bit mode.

| old | new | | |
|---|---|---|---|
| 245 | | – ADC0CN = 0x02; | // ADC0 disabled, normal tracking, |
| | 241 | + ADC0CN0 = 0x02; | // ADC0 disabled, |
| 246 | 242 | | // conversion triggered on TMR2 overflow |
| 247 | | – | |
| 248 | | – REF0CN = 0x03; | // Enable on-chip VREF and buffer |
| 249 | | – | |
| 250 | | – AMX0P = 0x0C; | // ADC0 positive input = P1.4 |
| 251 | | – AMX0N = 0x11; | // ADC0 negative input = GND |
| 252 | | – | // i.e., single ended mode |
| 253 | | – | |
| 254 | | – ADC0CF = ((SYSCLK/3000000)-1)<<3; | // set SAR clock to 3MHz |
| 255 | | – | |
| 256 | | – ADC0CF \|= 0x00; | // right-justify results |
| 257 | | – | |
| | 243 | + REF0CN = 0x08; | // References are the VDD and GND pins |
| | 244 | + ADC0AC = 0x00; | // 12-bit and accumulation disabled,right-justified, |
| | 245 | + | // repeat count set to 1 |
| | 246 | + ADC0PWR = 0x00; | // Low power mode disabled |
| | 247 | + ADC0MX = 0x0A; | // ADC0 positive input = P1.2 |
| | 248 | + ADC0CF = ((SYSCLK/12250000)-1)<<3; | // set SAR clock to 12.25MHz |
| | 249 | + ADC0CF \|= 0x01; | // 1x gain, normal tracking, 10-bit mode |
| 258 | 250 | EIE1 \|= 0x08; | // enable ADC0 conversion complete int. |
| 259 | 251 | | |
| 260 | | – AD0EN = 1; | // enable ADC0 |
| | 252 | + ADEN = 1; | // enable ADC0 |

8. Configure the ADC ISR with the 3.3 V reference voltage, ADINT bit name, and P1.2 in the printf output message.

| old | new | | |
|---|---|---|---|
| 325 | | – AD0INT = 0; | // clear ADC0 conv. complete flag |
| | 317 | + ADINT = 0; | // clear ADC0 conv. complete flag |
| 343 | | – mV = result * 2440 / 1023; | |
| 344 | | – printf("P1.4 voltage: %ld mV\n",mV); | |
| | 335 | + mV = result * 3300 / 1023; | |
| | 336 | + printf("P1.2 voltage: %ld mV\n",mV); | |

9. Update the transmit bit name in the UART0 initialization.

| old | new | | |
|---|---|---|---|
| 302 | | – TI0 = 1; | // Indicate TX0 ready |
| | 294 | + TI = 1; | // Indicate TX0 ready |

10. Perform final file cleanup, which includes removing unnecessary 16-bit SFR definitions from the top of the file. These are now defined in the **C8051F850_defs.h** file.

| old | new | | |
|---|---|---|---|
| 118 | | – -//--- | |
| 119 | | – -// 16-bit SFR Definitions for 'F33x | |
| 120 | | – -//--- | |
| 121 | 119 | | |
| 122 | | – sfr16 TMR2RL = 0xca; | // Timer2 reload value |
| 123 | | – sfr16 TMR2 = 0xcc; | // Timer2 counter |
| 124 | | – sfr16 ADC0 = 0xbd; | // ADC0 result |

11. Build the project and fix any errors. At this point, the code is ported and the equivalent of the original C8051F330/1/2/3/4/5 example. Running the code on a C8051F85x/86x Target Board (ensure all jumpers on J27, J7, and JP2 are populated) and rotating the potentiometer will change the output value. The UART output can be viewed using any terminal (RealTerm) program.
12. Now, add burst mode and auto-averaging to the example to take advantage of the new C8051F85x/86x features. Set repeat count to 64 conversions, setup the values of the ADC0PWR and AD0TK fields according to **Table 14.2 ADC0 Optimal Power Configuration (8 and 10-bit Mode)** in the C8051F85x/86x data sheet, and enable burst mode.

| old | new | | |
|---|---|---|---|
| 244 | | – ADC0AC = 0x00; | // 12-bit and accumulation disabled,right-justified, |
| 245 | | – | // repeat count set to 1 |
| | 244 | + ADC0AC = 0x05; | // 12-bit disabled,right-justified, |
| | 245 | + | // repeat count set to 64 conversions |
| 252 | | – ADEN = 1; | // enable ADC0 |
| | 252 | + ADC0PWR = 0x44; | |
| | 253 | + ADC0TK = 0x3A; | |
| | 254 | + ADBMEN = 1; | // Burst mode enabled |
| 325 | | – result = accumulator / 2048; | |
| | 327 | + result = accumulator / 2048 / 64; | |

13. Build the project and fix any errors. Running the code on a C8051F85x/86x Target Board (ensure all jumpers on J27, J7, and JP2 are populated) and rotating the potentiometer will change the output value. The UART output can be viewed using any terminal (RealTerm) program.

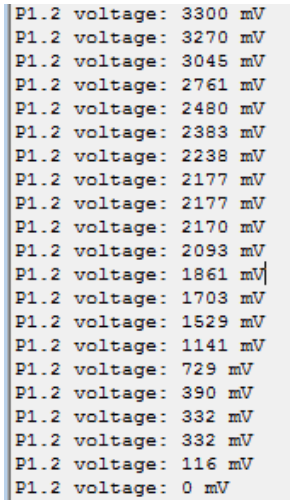
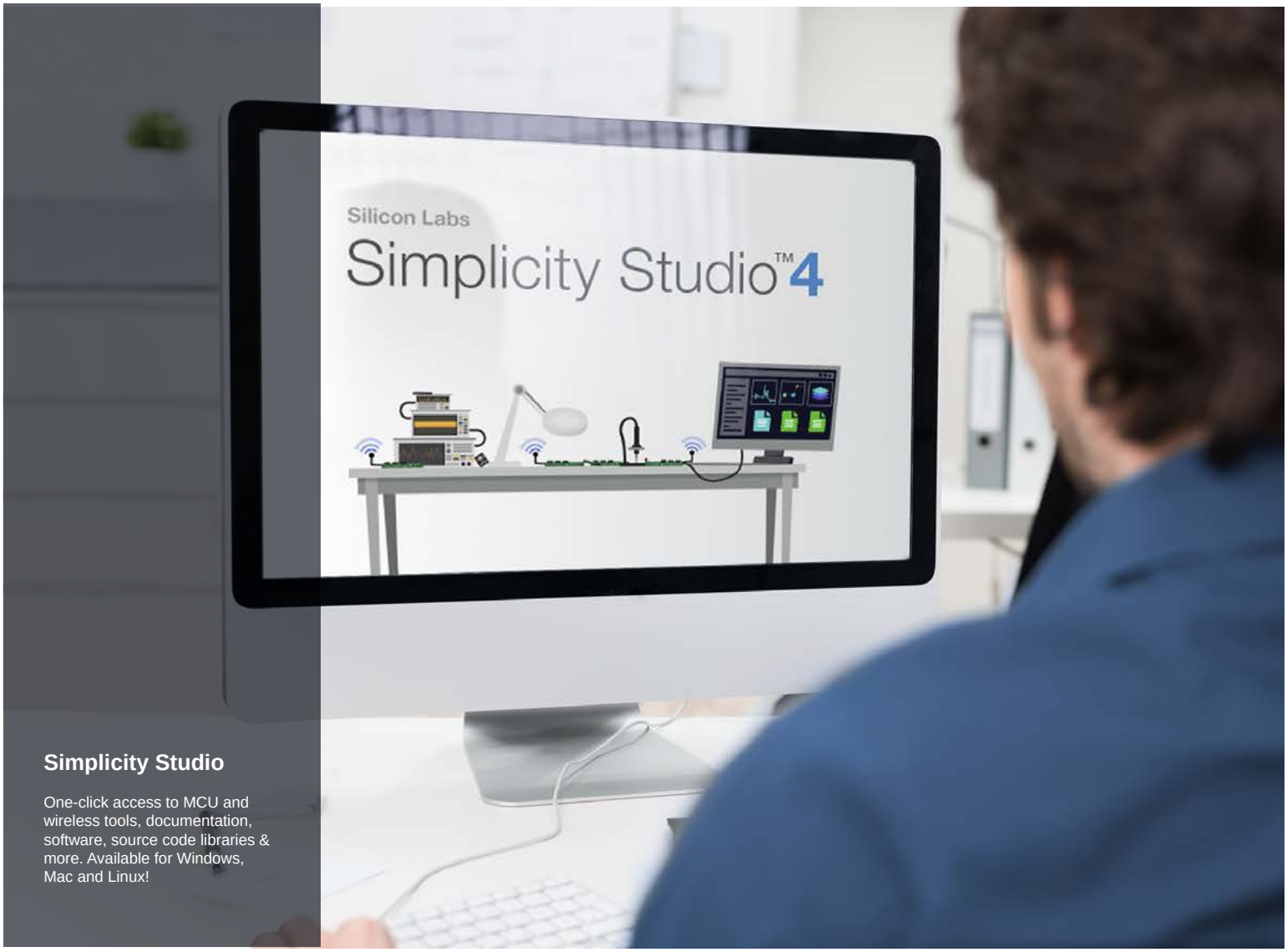



**Figure 2. ADC Porting Example UART0 Output**

The original **F33x_ADC0_ExternalInput.c** file and modified **F85x_ADC0_porting_example.c** file are included with this document as an example of this porting process. The firmware package can be downloaded from www.siliconlabs.com/8bit-appnotes.




## Simplicity Studio

One-click access to MCU and wireless tools, documentation, software, source code libraries & more. Available for Windows, Mac and Linux!

**IoT Portfolio**
*www.silabs.com/IoT*

**SW/HW**
*www.silabs.com/simplicity*

**Quality**
*www.silabs.com/quality*

**Support and Community**
*community.silabs.com*



SILICON LABS

**Silicon Laboratories Inc.**
**400 West Cesar Chavez**
**Austin, TX 78701**
**USA**

**http://www.silabs.com**