SILICON LABS

www.silabs.com

C8051T6xx/3xx One Time Programmable (OTP) USB MCUs

Agenda

- C8051T6xx/3xx family overview
- C8051T6xx/3xx family differences from flash-based devices
- OTP development flow
- Other considerations
- Development tools
- Summary

SILICON LABS

www.silabs.com

USB OTP Device Family

Introducing the C8051T62x/32x

- **Reduce cost, simplify design and shorten development time**
  - USB crystal-less operation capability
  - Best-in-class analog capabilities five times faster than any competitor
  - Accomplish more work in less time with a high performance processing core
  - OTP versions for very cost-sensitive applications

- **Accelerate time-to-market**
  - Production-ready software drivers
  - Step-by-step application notes and code examples
  - Easy-to-use development tools

- **Pin and code compatible enabling an easy migration path**
  - OTP C8051T32x is compatible with Flash-based C8051F32x enabling a cost reduction path

SILICON LABS

USB Design Challenges

- **Typical USB microcontrollers lack high-precision analog capabilities creating a more complex and expensive system solution**
  - Higher BOM cost: external components are required
  - Significant hardware and software design effort

- **Most applications require more than just USB connectivity**
  - Most MCUs are designed to enable only USB connectivity
  - Multi-tasking operation can quickly saturate CPU performance

- **Competitive limitations**
  - External analog components are required increasing BOM cost and complexity
  - Expensive high-end or chip set solutions are implemented to overcome performance bottleneck

SILICON LABS

www.silabs.com

OTP and Flash Device Differences
C8051T62x/32x vs. C8051F34A

Code Memory Storage

- **Flash memory used on C8051F34A family**
- **Byte-programmable EPROM code memory on the C8051T62x/32x families**
  - When pre-fetch engine is enabled (default) timing is similar to flash devices
  - Porting considerations
    - Insure no firmware routines exist to erase code memory
    - In application firmware can write to memory, but only once
  - Add a 4.7 uF capacitor to enable programming the V_{PP} pin to ground
    - C8051T62x and C8051T32x devices

Feature	C8051F34A	C8051T62x/32x
Code memory can be erased and reprogrammed	Yes	No
Programming voltage (V_{PP}) required to program code memory	No	Yes
Code memory can be erased from firmware on the device	Yes	No
Code memory can be written from firmware on the device	Yes	
Code memory can be read from firmware on the device	Yes	

Special Function Registers (SFR) (1 of 2)

- **Differences related to functionality and features**
- **SFRs can exist in one family and not another**
  - Reading and writing these registers does not cause any problems if not present
  - Porting considerations
    - None
  - Example: P3 register is not found in the C8051T622 and is on the F34A

F8	SPI0CN	PCA0L	PCA0H	PCA0CPL0	PCA0CPH0	PCA0CPL4	PCA0CPH4	VDM0CN
F0	B	P0MDIN	P1MDIN	P2MDIN	PCA0PWM (P3MDIN)	IAPCN (P4MDIN)	EIP1	EIP2
E8	ADC0CN	PCA0CPL1	PCA0CPH1	PCA0CPL2	PCA0CPH2	PCA0CPL3	PCA0CPH3	RSTSRC
E0	ACC	XBR0	XBR1	XBR2	IT01CF	SMOD1	EIE1	EIE2
D8	PCA0CN	PCA0MD	PCA0CPM0	PCA0CPM1	PCA0CPM2	PCA0CPM3	PCA0CPM4	P3SKIP
D0	PSW	REF0CN	SCON1	SBUF1	P0SKIP	P1SKIP	P2SKIP	USB0XCN
C8	TMR2CN	REG01CN (REG0CN)	TMR2RLL	TMR2RLH	TMR2L	TMR2H	-	SMB0ADM (-)
C0	SMB0CN	SMB0CF	SMB0DAT	ADC0GTL	ADC0GTH	ADC0LTL	ADC0LTH	SMB0ADR (P4)
B8	IP	CLKMUL	P1MASK (AMX0N)	AMX0P	ADC0CF	ADC0L	ADC0H	-
B0	P3	OSCXCN	OSCICN	OSCICL	SBRLL1	SBRLH1	P1MAT (FLSCL)	MEMKEY (FLKEY)
A8	IE	CLKSEL	EMI0CN		SBCON1	-	P0MASK (P4MDOUT)	PFE0CN
A0	P2				OUT	P1MDOUT	P2MDOUT	P3MDOUT
98	SCON0				MD	CPT0MD	CPT1MX	CPT0MX
90	P1				3L	TMR3H	USB0ADR	USB0DAT
88	TCON				0	TH1	CKCON	PSCTL
80	P0	SP	DPL	DPH	P0MAT (EMI0TC)	EMI0CF	OSCLCN	PCON
	0(8) Bit-Addressable	1(9)	2(A)	3(B)	4(C)	5(D)	6(E)	7(F)

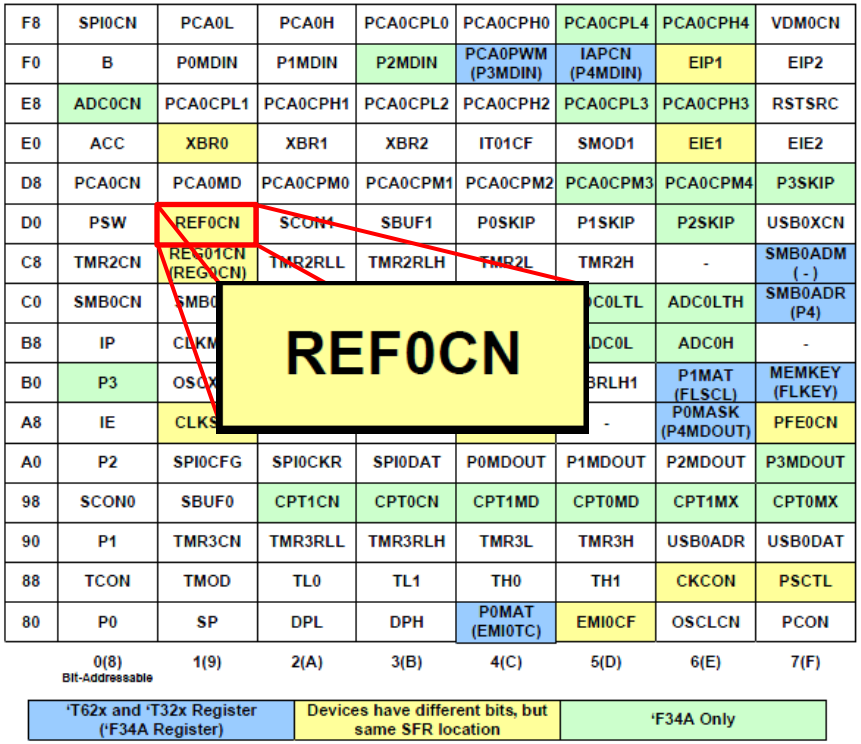
P3

'T62x and 'T32x Register ('F34A Register) | Devices have different bits, but same SFR location | 'F34A Only

SILICON LABS

Special Function Registers (SFR) (2 of 2)

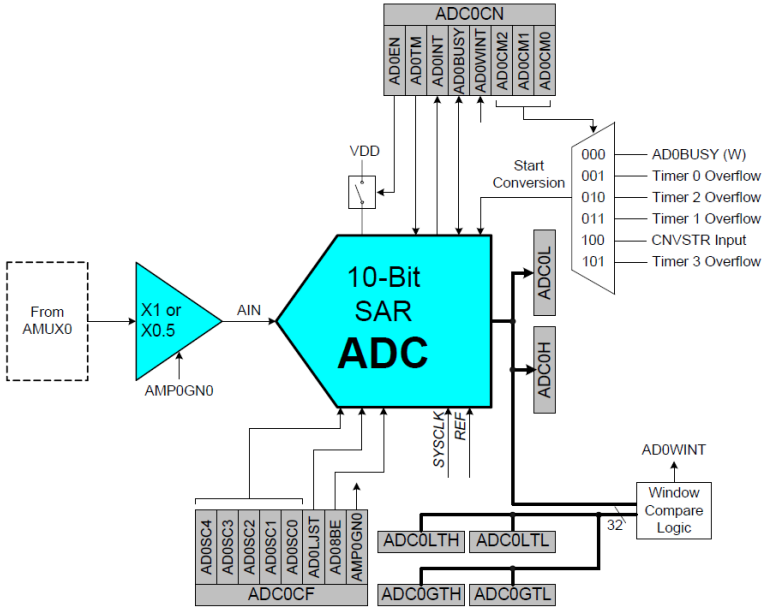
- **Some registers have additional bits defined**
  - Peripheral behavior remains unchanged if the default settings are used
  - Porting considerations
    - To maintain functionality verify that default bit settings are used for additional bits in common registers
  - Example:
    - REF0CN register adds REFBGS to halve the ADC reference voltage
    - Default setting maintains functionality with the C8051F34A



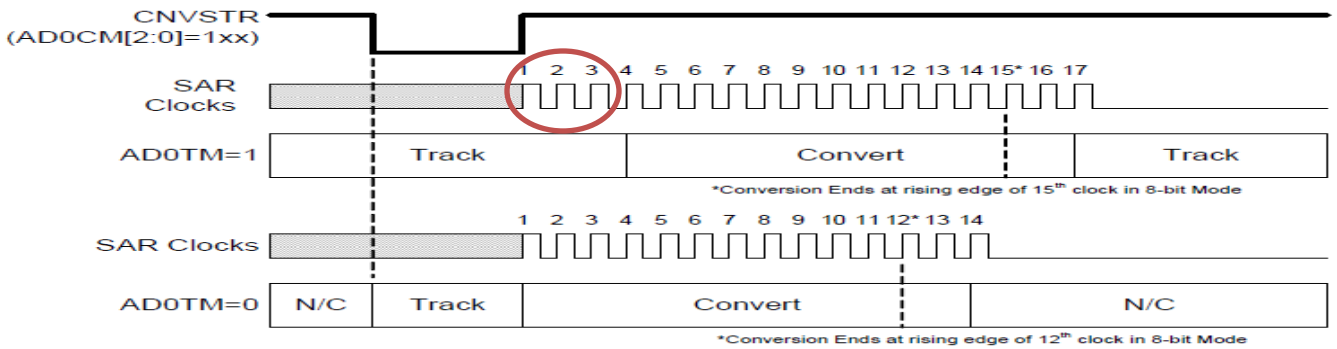
SILICON LABS

Analog Considerations

- **ADC sample rate increase to 500 ksps**
  - SAR clock increased to 8.33 MHz
- **Gain setting of 0.5x now available**
- **Single ended inputs only**
- **External conversion start timing provides additional options**



ADC Diagram

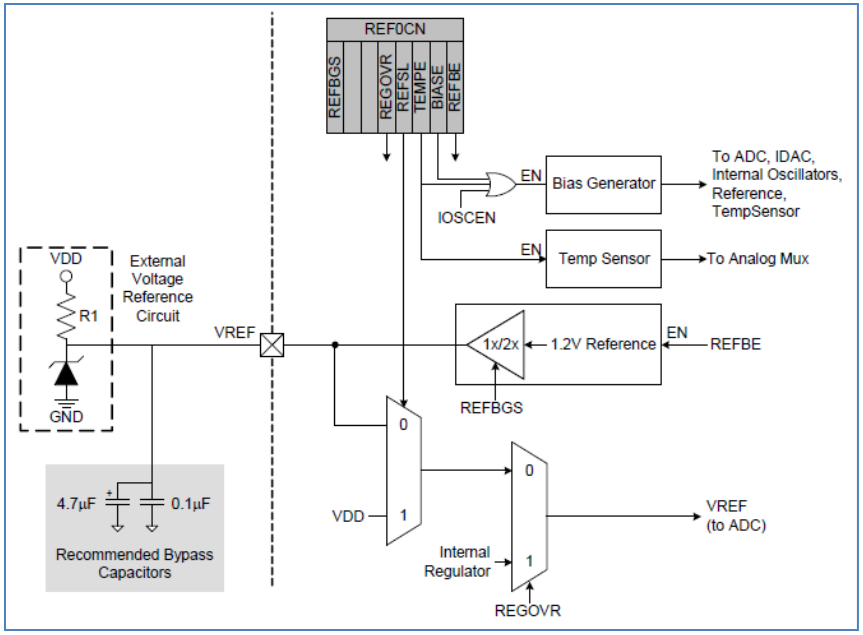


ADC CNVSTR Timing

SILICON LABS

Analog Considerations

- **More voltage reference options**
- **Calibrated temperature sensor**
- **Porting considerations**
  - AMX0CN register should always be written as 11111b
  - Default register settings for the reference selection maintain functionality
  - Temperature sensors have different transfer functions and firmware should be adjusted accordingly



VREF Diagram

Supply Voltage Considerations (1 of 2)

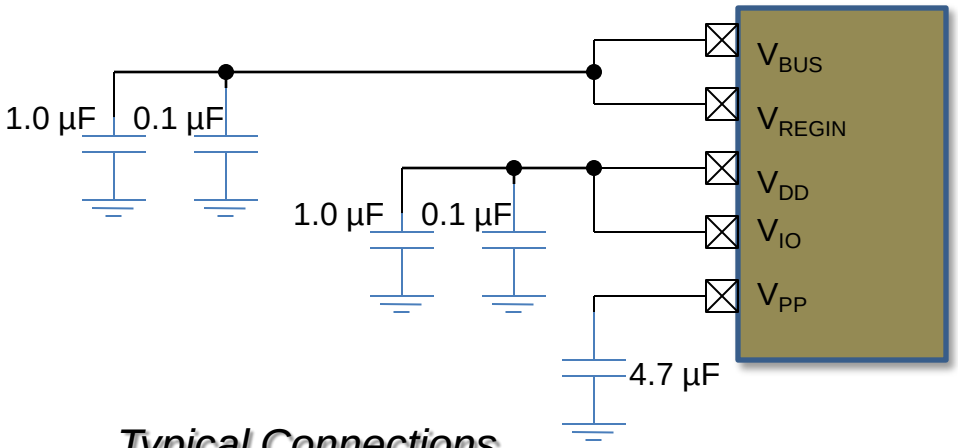
- **Process technology change and second voltage regulator added**
  - V_{DD} output now 3.45 V instead of 3.3 V
  - Second regulator provides 1.8 V
    - Additional registers to support the regulator functionality (REG01CN)
    - Can be placed in a low power mode
  - V_{IO} pin added on some devices in case the port input/output voltages are required to be different from the V_{DD} that the device is operating
  - V_{DD} monitor threshold voltage changes
  - Porting considerations
    - None for firmware, but care must be observed for electrical connections

Feature	C8051F34A	C8051T63x/32x
Supply voltage range	2.7–3.6 V	1.8–3.6 V
3.3 V regulator for V_{DD}	Yes	No
3.45 V regulator for V_{DD}	No	Yes
1.8 V regulator for internal core voltage	No	Yes
Maximum voltage on any I/O pin	5.8 V	*V_{DD} + 3.6 V (5.8V max)

*If supply voltage reduced to 0 V then voltage at the pin must be less than 3.6 V

Supply Voltage Considerations (2 of 2)

- **V_{IO} considerations**
  - $V_{IO} \leq V_{DD}$
  - Not all packages have a V_{IO} pin
  - Reset can be pulled up to VDD
- **V_{PP} considerations**
  - When using in-application programming (IAP) a 4.7 uF capacitor is required on the V_{PP} pin
  - It is not recommended to use the V_{PP} pin as GPIO if IAP to be used
    - If GPIO and IAP are required then the external circuit on the pin must not provide a load when the programming is enabled



Typical Connections

SILICON LABS

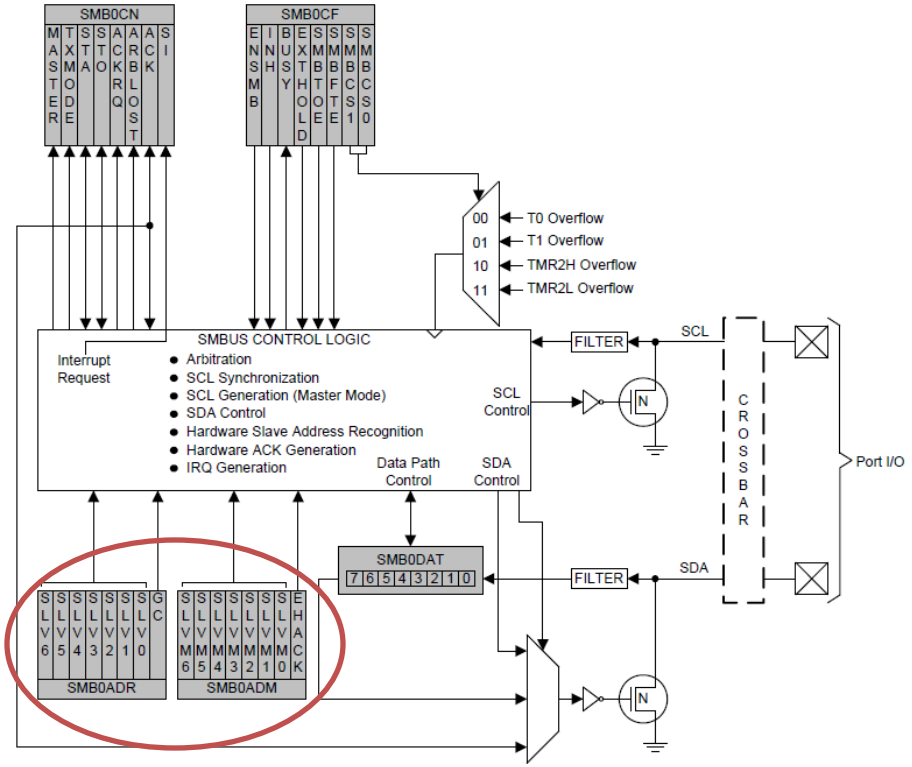
Low Power Modes and Clocking

- **Suspend mode operation turns off the internal oscillator**
  - C8051F34A requires USB resume signaling or VBUS interrupt to exit suspend
  - C8051T62x/32x devices exit suspend using
    - Resume signaling or VBUS interrupt
    - Port match
    - Timer 3 if running from external oscillator or the low frequency internal oscillator
- **Clocking options vary between devices**
  - Porting considerations
    - CLKMUL register remains across all devices for compatibility even though the internal oscillator is used to drive the USBCLK directly

Feature	C8051F34A	C8051T62x
Internal calibrated 24.5 MHz oscillator (divided by 1, 2, 4 or 8)	Yes	No
Internal calibrated 48 MHz oscillator (divided by 1,2,4 or 8)	No	Yes
Internal 80 kHz oscillator (divided by 1, 2, 4 or 8)	Yes	Yes
External CMOS clock (digital input)	Yes	Yes
External oscillator in RC or capacitor mode	Yes	Yes
External oscillator in crystal oscillator mode	Yes	Yes

Additional Features

- **SMBus/I²C**
  - Optional hardware address recognition and automatic ACK
    - Reduces firmware overhead
- **Port match**
  - Allows system events to be triggered by a logic value change on a port pin
  - Can generate interrupts
  - Can wake the device from suspend mode
- **PCA**
  - Includes 9, 10 and 11 bit PWM generation



SILICON LABS

SILICON LABS

www.silabs.com

Developing USB OTP Applications

The C8051T62x/32x Development Kit

- **Kit contents for C8051T620 and C8051T622**
  - C8051T62x motherboard
  - C8051T62x emulation daughter board with C8051F34A installed
  - Socket daughter board (one of the following):
    - C8051T62x QFN 32-pin (C8051T620DK)
    - C8051T622 QFN 24-pin (C8051T622DK)
  - Twenty device samples (one of the following):
    - C8051T620-GM (C8051T620DK)
    - C8051T622-GM (C8051T622DK)
  - C8051Txxx development kit quick-start guide
  - Product information CD-ROM includes:
    - Silicon Labs Integrated Development Environment (IDE)
    - Evaluation version of 8051 development tools (macro assembler, linker, C compiler)
    - Source code examples and register definition files
    - Documentation
  - AC-to-DC universal power adapter
  - Two USB cables

SILICON LABS

Required Software

- **Required software**
  - Silicon Labs IDE or 3rd party IDE
  - C compiler—code limited evaluation versions supplied with the kit
- **Recommended software**
  - Configuration wizard—Configuration Wizard 2
  - Virtual com port (VCP) drivers
  - ToolStick Terminal
  - uVision driver for Keil if using the uVision IDE

Software can be downloaded at
http://www.silabs.com/mcudownloads

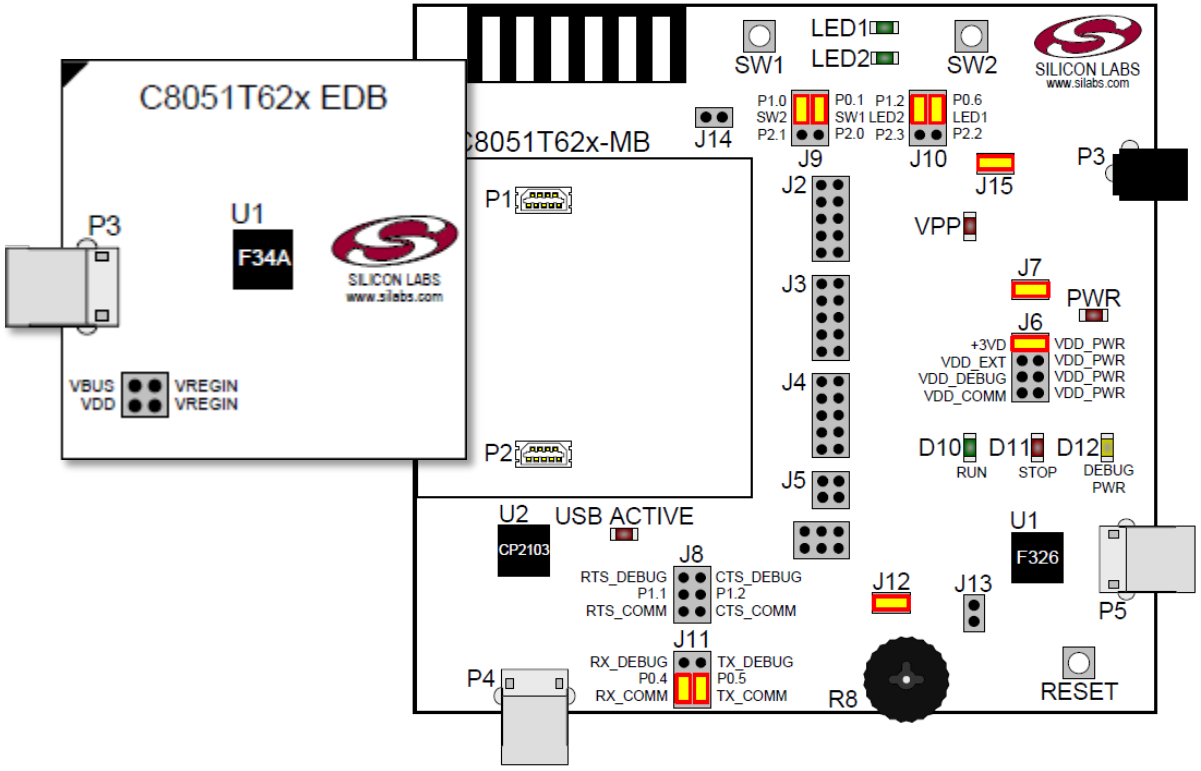
SILICON LABS

www.silabs.com

Using the Kits

Attaching a Daughter Card

- **Development can start using the flash-based C8051F34A**
  - Plug the C8051T62x EDB emulation daughter board into the motherboard sockets P1 and P2 (C8051T62x EDB has the C8051F34A device)

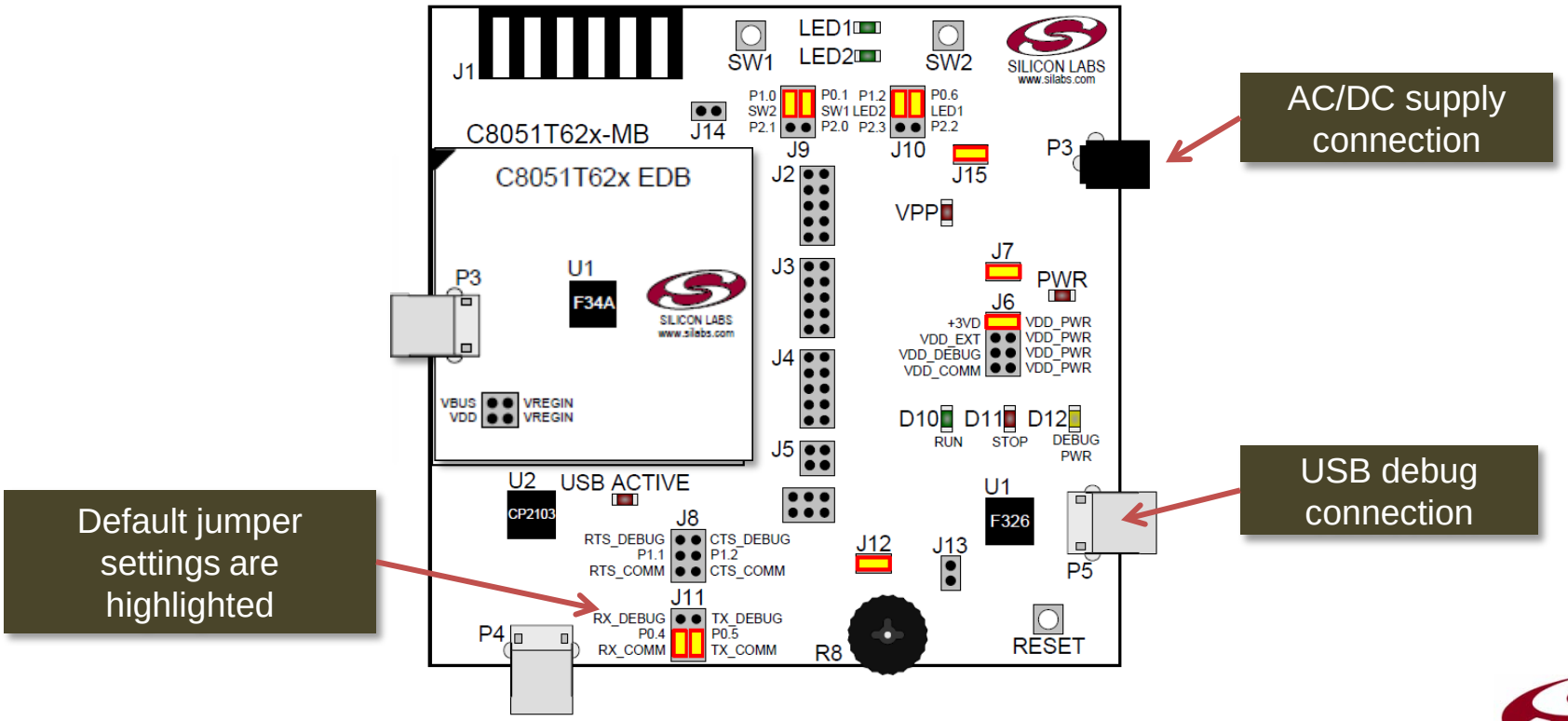


Using the C8051F34A for Development

SILICON LABS

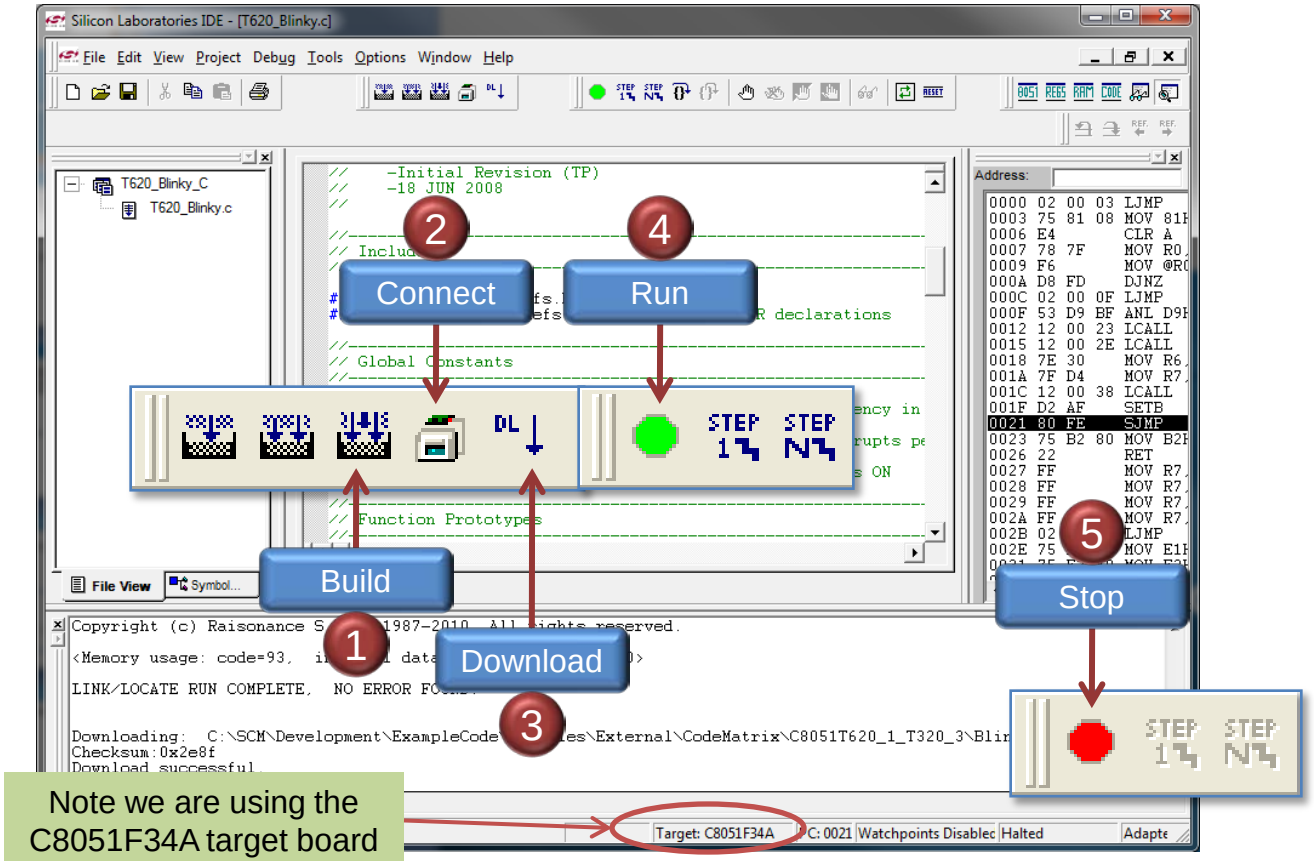
Making Mother Board Connections

- **Verify jumper settings with the DK user's guide**
- **Connect USB cable to the mother board P5**
  - Provides code download and debug capability
  - Provides interface to targets UART peripheral if enabled using J11
- **Connect the AC/DC power adapter to the barrel plug P3**



Verify Tool Flow

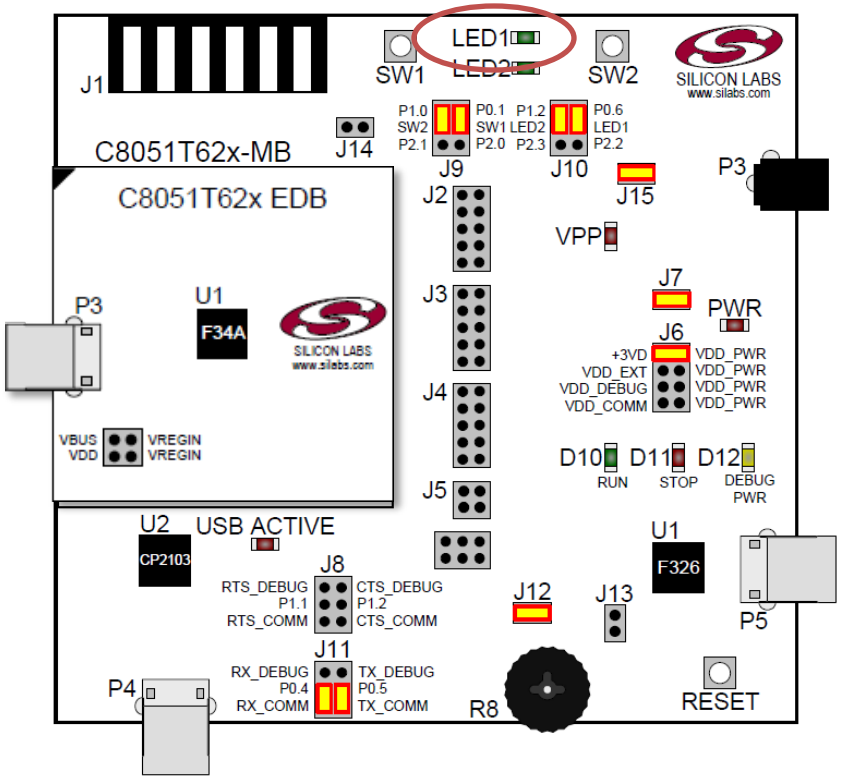
- **Build a sample project**
  - Open T620_Blinky_C.wsp project using the Silicon Labs IDE
    - Found in the C:\Silabs\MCU\Examples\C8051T620_1_T320_3 directory
  - Build, connect, download and run the project



Silicon Labs IDE

Blinking the LED

- **When the application is running LED1 should be blinking**
- **Code can be modified and downloaded multiple times using the C8051F34A**

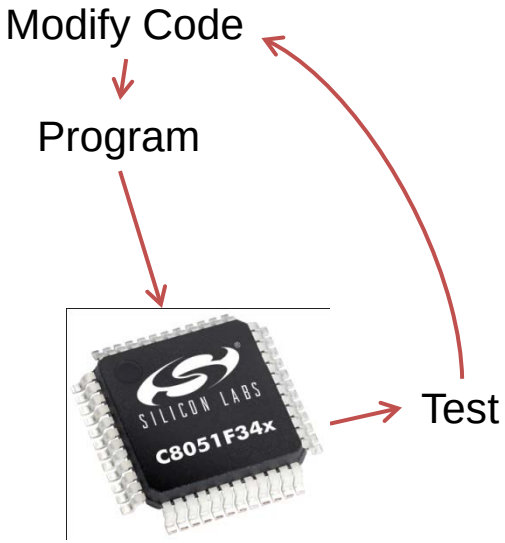


Running the Test Application

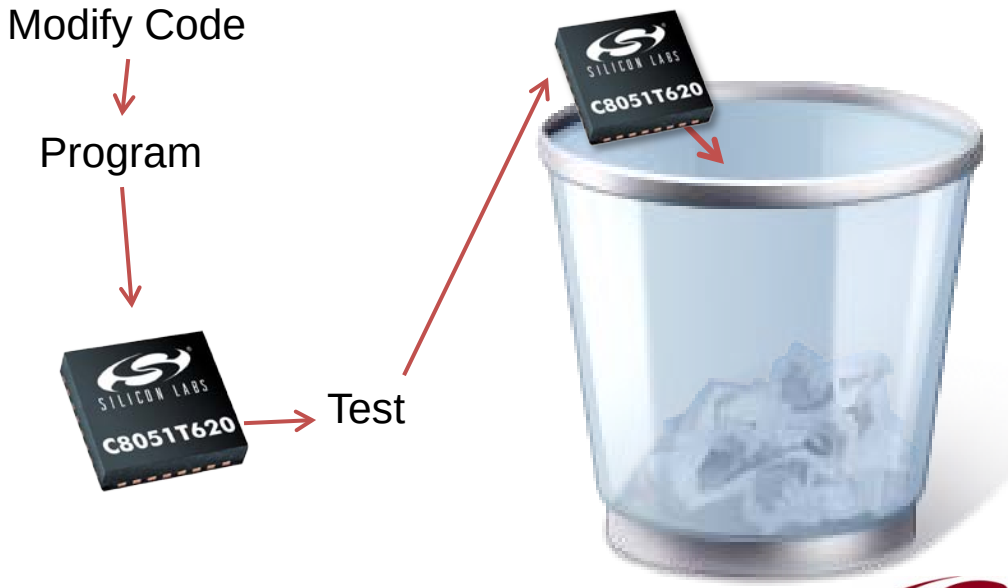
SILICON LABS

Developing the Application

- **Make modifications to the example code to provide the required system functionality (recommended) or write the application from scratch**
- **Using the flash-based C8051F34A many code iterations can be done without having to burn the code into the OTP device**
  - Since OTP devices can only be programmed once they would have to be discarded after each code test



Flash Code Flow



OTP Code Flow

Porting the Application to the OTP Device

- **Once the application code has been completed on the flash-based MCU migrate the project to the OTP version**
  - Make necessary porting changes based on MCU differences
  - USB clock recovery step size

```
//-----------------------------------------------------------------------
// USB0_Init
//-----------------------------------------------------------------------
//
// Return Value - None
// Parameters - None
//
// - Initialize USB0
// - Enable USB0 interrupts
// - Enable USB0 transceiver
// - Enable USB0 with suspend detection
//
// -----------------------------------------------------------------------
void Usb_Init(void)
{
   POLL_WRITE_BYTE(POWER,  0x08); // Force Asynchronous USB Reset
   POLL_WRITE_BYTE(IN1IE,  0x07); // Enable Endpoint 0-2 in interrupts
   POLL_WRITE_BYTE(OUT1IE, 0x07); // Enable Endpoint 0-2 out interrupts
   POLL_WRITE_BYTE(CMIE,   0x07); // Enable Reset, Resume, and Suspend
                                  // interrupts
#ifdef _USB_LOW_SPEED_
   USB0XCN = 0xC0;                // Enable transceiver; select low speed
   POLL_WRITE_BYTE(CLKREC, 0xA9); // Enable clock recovery; single-step
                                  // mode disabled; low speed mode enabled
#else
   USB0XCN = 0xE0;                // Enable transceiver; select full speed
   POLL_WRITE_BYTE(CLKREC, 0x89); // Enable clock recovery, single-step
                                  // mode disabled
#endif /* _USB_LOW_SPEED_ */

   EIE1 |= 0x02;                  // Enable USB0 Interrupts
   EA = 1;                        // Global Interrupt enable
                                  // Enable USB0 by clearing the USB
                                  // Inhibit bit
   POLL_WRITE_BYTE(POWER,  0x01); // and enable suspend detection
}
```

Flash Code Flow

```
//-----------------------------------------------------------------------
// USB0_Init
//-----------------------------------------------------------------------
//
// Return Value - None
// Parameters - None
//
// - Initialize USB0
// - Enable USB0 interrupts
// - Enable USB0 transceiver
// - Enable USB0 with suspend detection
//
// -----------------------------------------------------------------------
void Usb_Init(void)
{
   POLL_WRITE_BYTE(POWER,  0x08); // Force Asynchronous USB Reset
   POLL_WRITE_BYTE(IN1IE,  0x07); // Enable Endpoint 0-2 in interrupts
   POLL_WRITE_BYTE(OUT1IE, 0x07); // Enable Endpoint 0-2 out interrupts
   POLL_WRITE_BYTE(CMIE,   0x07); // Enable Reset, Resume, and Suspend
                                  // interrupts
#ifdef _USB_LOW_SPEED_
   USB0XCN = 0xC0;                // Enable transceiver; select low speed
   POLL_WRITE_BYTE(CLKREC, 0xAF); // Enable clock recovery; single-step
                                  // mode disabled; low speed mode enabled
#else
   USB0XCN = 0xE0;                // Enable transceiver; select full speed
   POLL_WRITE_BYTE(CLKREC, 0x8F); // Enable clock recovery, single-step
                                  // mode disabled
#endif /* _USB_LOW_SPEED_ */

   EIE1 |= 0x02;                  // Enable USB0 Interrupts
   EA = 1;                        // Global Interrupt enable
                                  // Enable USB0 by clearing the USB
                                  // Inhibit bit
   POLL_WRITE_BYTE(POWER,  0x01); // and enable suspend detection
}
```

OTP Code Flow

Clock Recovery Port Example

SILICON LABS

Using the Oscillator

- **Internal oscillator is now 48 MHz instead of 12 MHz**
- **For backward compatibility the clock multiplier registers remain although they provide no functionality**

```
//-----------------------------------------------------------------------------
// Sysclk_Init
//-----------------------------------------------------------------------------
//
// Return Value - None
// Parameters - None
//
// Initialize system clock to maximum frequency.
//
// -----------------------------------------------------------------------------
void Sysclk_Init(void)
{
#ifdef _USB_LOW_SPEED_

   OSCICN |= 0x03;                     // Configure internal oscillator for
                                       // its maximum frequency and enable
                                       // missing clock detector

   CLKSEL  = SYS_INT_OSC;              // Select System clock
   CLKSEL |= USB_INT_OSC_DIV_2;        // Select USB clock
#else
   OSCICN |= 0x03;                     // Configure internal oscillator for
                                       // its maximum frequency and enable
                                       // missing clock detector

   // This clock multiplier code is no longer necessary, but it is retained
   // here for backwards compatibility with the 'F34x.

   CLKMUL  = 0x00;                     // Select internal oscillator as
                                       // input to clock multiplier

   CLKMUL |= 0x80;                     // Enable clock multiplier
   Delay();                            // Delay for clock multiplier to begin
   CLKMUL |= 0xC0;                     // Initialize the clock multiplier
   Delay();                            // Delay for clock multiplier to begin

   while(!(CLKMUL & 0x20));            // Wait for multiplier to lock
   CLKSEL  = SYS_INT_OSC;              // Select system clock
   CLKSEL |= USB_4X_CLOCK;             // Select USB clock
#endif  /* _USB_LOW_SPEED_ */
}
```

Code remains from the C8051F34A. It can be removed when using the C8051T62x.

Applications with an ADC

- **Voltage reference options can be optimized for dynamic range**
- **C8051T62x/32x is single ended and doesn't have a mux for the negative input**
- **SAR clock can remain the same or can be increased for faster sample rates**

```
//-----------------------------------------------------------------------------
// ADC0_Init
//-----------------------------------------------------------------------------
//
// Return Value:  None
// Parameters:    None
//
// Configures ADC0 to make single-ended analog measurements on pin P1.1
//
//-----------------------------------------------------------------------------

void ADC0_Init (void)
{
   ADC0CN = 0x02;                      // ADC0 disabled, normal tracking,
                                       // conversion triggered on TMR2 overflow

   REF0CN = 0x03;                      // Enable on-chip VREF and buffer

   AMX0P = 0x13;                       // ADC0 positive input = P1.1
   AMX0N = 0x1F;                       // ADC0 negative input = GND
                                       // i.e., single ended mode

   ADC0CF = ((SYSCLK/3000000)-1)<<3;   // set SAR clock to 3MHz

   ADC0CF |= 0x00;                     // right-justify results

   EIE1 |= 0x08;                       // enable ADC0 conversion complete int.

   AD0EN = 1;                          // enable ADC0
}
```

Flash Code Flow

```
//-----------------------------------------------------------------------------
// ADC0_Init
//-----------------------------------------------------------------------------
//
// Return Value:  None
// Parameters:    None
//
// Configures ADC0 to make single-ended analog measurements on pin P2.5.
//
//-----------------------------------------------------------------------------
void ADC0_Init (void)
{
   ADC0CN = 0x02;                      // ADC0 disabled, normal tracking,
                                       // conversion triggered on TMR2 overflow

   REF0CN = 0x03;                      // Enable on-chip VREF and buffer

   AMX0P = 0x0D;                       // ADC0 positive input = P2.5

   ADC0CF = ((SYSCLK/3000000)-1)<<3;   // Set SAR clock to 3MHz

   ADC0CF |= 0x00;                     // Right-justify results
   ADC0CF |= 0x01;                     // Gain = 1

   EIE1 |= 0x08;                       // Enable ADC0 conversion complete int.

   AD0EN = 1;                          // Enable ADC0
}
```

OTP Code Flow

ADC Example

SILICON LABS

Measuring Temperature

- **Temperature sensor measurements differ between the two families**
  - Transfer function of the temperature sensors is different
- **OTP devices have temperature compensation at 0 ˚C using V_{DD}**

```
#define COMP_ADDRESS     0x3FFA          // Location of TOFFH and TOFFL

U16 code COMPENSATION _at_ COMP_ADDRESS; // TOFFH and TOFFL stored in EPROM
                                         // memory
```

•Compensation value stored in code memory

```
//-----------------------------------------------------------------------------
// ADC0_Init
//-----------------------------------------------------------------------------
//
// Return Value : None
// Parameters   : None
//
// Initialize the ADC to use the temperature sensor.
//
//-----------------------------------------------------------------------------
void ADC0_Init (void)
{
   REF0CN = 0x0E;                      // VREF is VDD, Temp. Sensor ON, Bias ON
   AMX0P = 0x1E;                       // Selects Temp. Sensor
   ADC0CF = ((SYSCLK/3000000)-1)<<3;   // Set SAR clock to 3MHz
   ADC0CF |= 0x04;                     // ADC0 is left justified

   ADC0CN = 0x82;                      // ADC ON, starts on TMR2 overflow

   EIE1 |= 0x08;                       // Enable ADC0 conversion complete int.
}
```

▪ADC uses V_{DD} as V_{REF}
▪Mux input set to temp sensor

```
temp_scaled *= SLOPE;          // Calculate rounded temperature

// With a left-justified ADC, we have to shift the decimal place
// of temp_scaled to the right so we can match the format of
// OFFSET. Once the formats are matched, we can subtract OFFSET.
temp_scaled = temp_scaled >> OVER_ROUND;

temp_scaled -= OFFSET;         // Apply offset to temp
temp_comp = temp_scaled - COMPENSATION; // Apply TOFFH and TOFFL
```

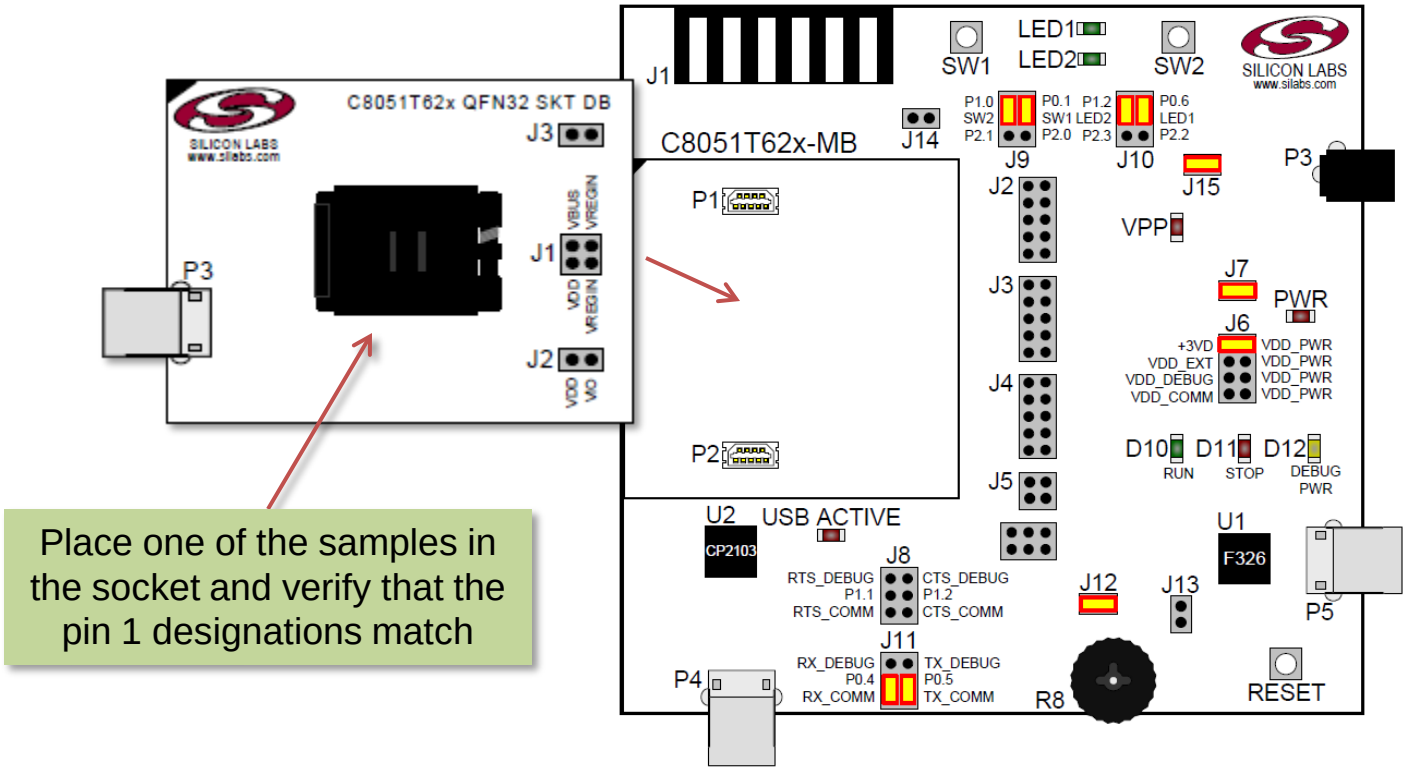
•Firmware uses the new slope, offset and compensation to determine temperature

OTP Code Flow

SILICON LABS

Change the Daughter Card

- **Once code porting has been completed**
  - Attach the C8051T620 SKT DB daughter card into the motherboard sockets P1 and P2
    - C8051T62x QFN SKT DN has a socket for the specific device package
    - Sample devices provided in the kit

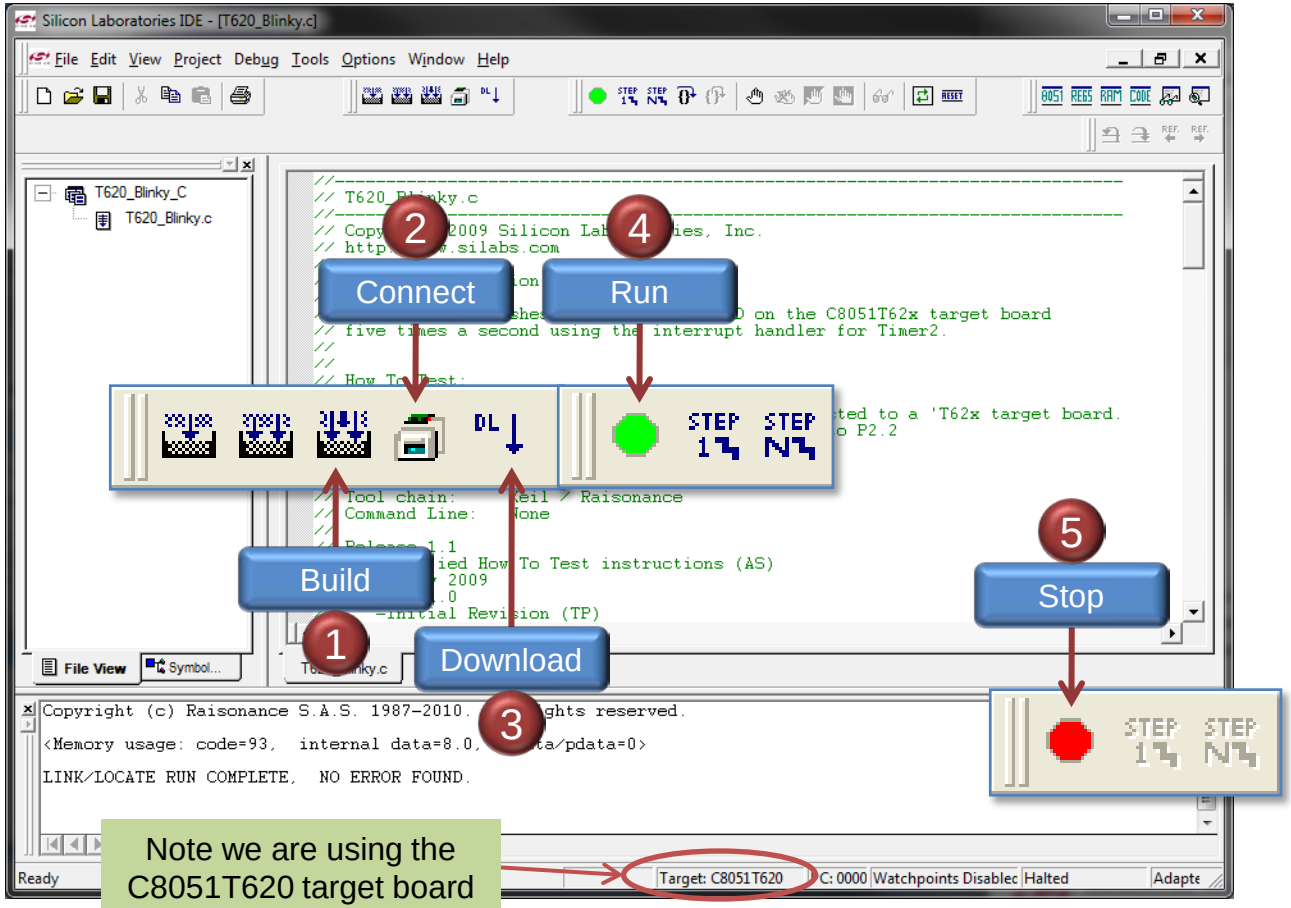


Attaching the C8051T62x Daughter Card

SILICON LABS

Build, Download and Run the Application

- **Test the OTP application**
  - Build, connect, download and run the project



Silicon Labs IDE

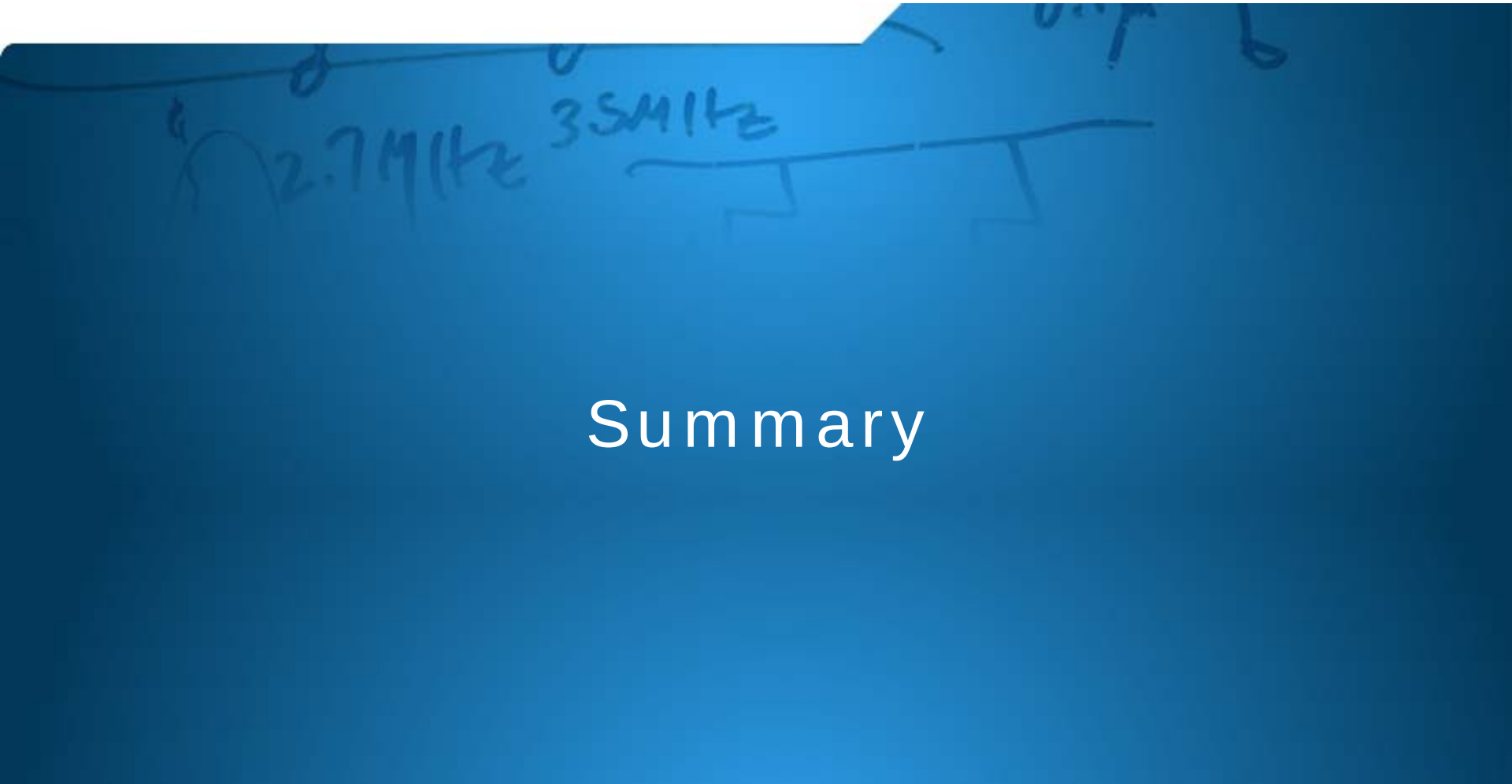
Available Documentation and Software

- Product data sheets available (www.silabs.com/USB)
- Data shorts available (www.silabs.com/USB)
- Example code included on IDE installation (www.silabs.com/MCUdownloads)
- USBXpress drivers (www.silabs.com/USBXpress)
- Application Notes available (www.silabs.com/USB)
  - **AN169** *USBXpress programmer's guide*
  - **AN200** *USB boot loader with shared USBXpress library*
  - **AN220** *USB driver customization*
  - **AN249** *Human interface device tutorial*
  - **AN368** *Difference between the C8051F34A and the C8051T62x and C8051T32x device families*
  - **AN455** *Porting code for C8051F320/1 devices to C8051T320/1 devices*
  - **AN456** *Porting code for C8051F326/7 devices to C8051T326/7 devices*
  - **AN456** *Porting code for C8051F326/7 to C8051T326/7 devices*
  - **AN532** *HID library API specification*

- Best-in-class product support and comprehensive software ecosystem
  - Silicon Labs offers free vendor PID (www.silabs.com/products/mcu/Pages/request-PID.aspx)
  - Pre-programming services

SILICON LABS

www.silabs.com



Summary

Summary

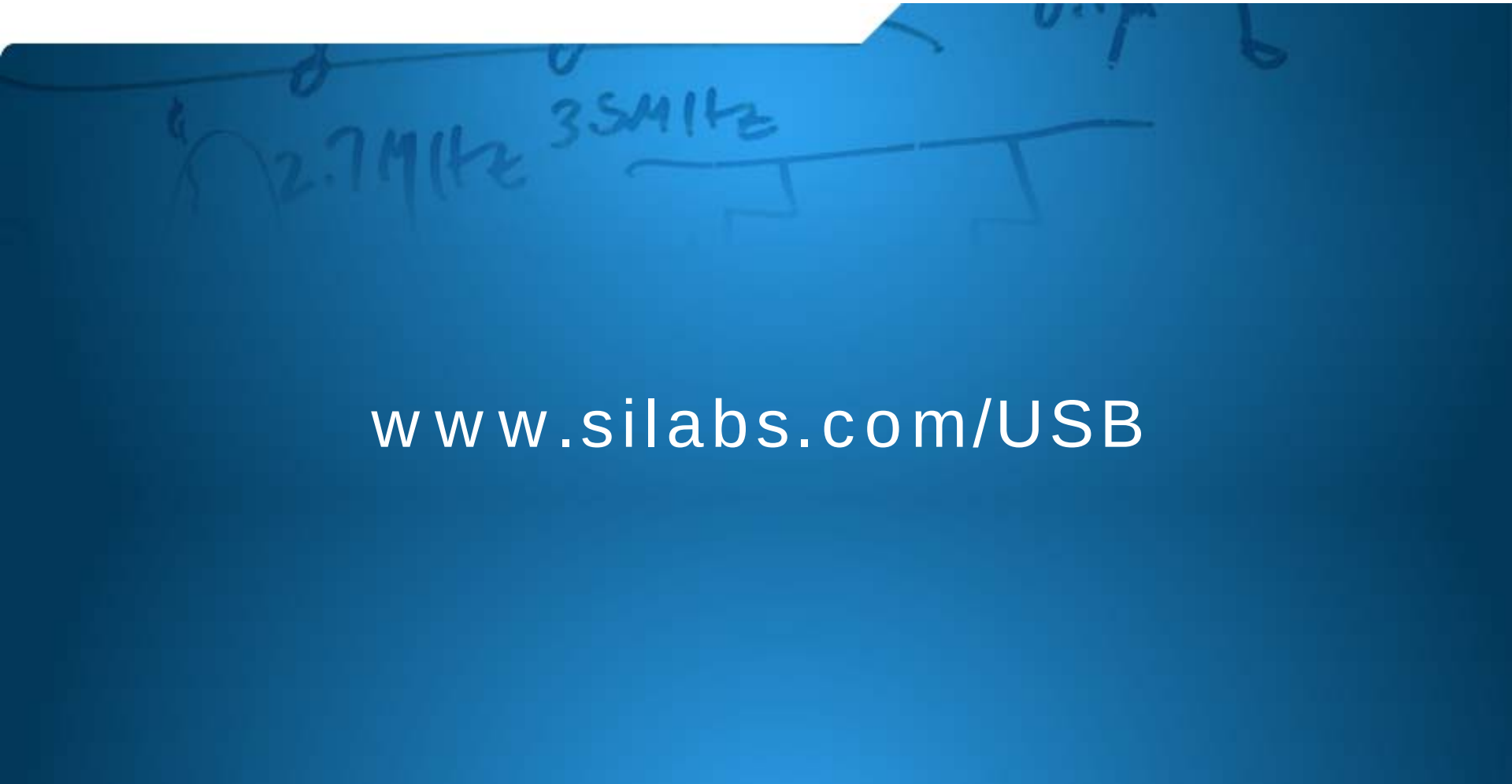
- **Silicon Labs USB solutions are designed to reduce cost, simplify design and shorten development time**
  - Best-in-class analog capabilities five times faster than any competitor
  - Accomplish more work in less time with a high performance processing core
  - USB crystal-less operation capability
  - OTP versions for cost sensitive applications

- **22 new products supported by a comprehensive development ecosystem**
  - Production-ready software
  - Step-by-step application notes and code examples
  - Easy-to-learn development tools

- **Pin and code compatible enabling an easy migration path**
  - C8051F38x is pin and code compatible with the C8051F34x
  - OTP C8051T32x is compatible with flash-based C8051F32x enabling a cost reduction path

SILICON LABS

www.silabs.com



www.silabs.com/USB

Mouser Electronics

Authorized Distributor

Click to View Pricing, Inventory, Delivery & Lifecycle Information:

Silicon Laboratories:

C8051T620DB32 C8051T620DK C8051T622DK