Application Note:
Modifying the MCU in Zigbee Tracker v2.0

1 Introduction

The Zigbee Tracker v2.0 utilizes a general-purpose STM32F1 series Microcontroller Unit (MCU). However, the User may require a different MCU (i.e. STM32L series for low power, ATMEGA series for better open source documentation/libraries) either in the same footprint or different, with a redesign of the Printed Circuit Board (PCB).
This document provides a description of the minimum specifications needed for the replacement MCU, the basic connections required to the other peripherals and the programming/debugging interface, to allow for easy integration of a different MCU into the Zigbee Tracker v2.0.

2 Integration of MCU

This Section covers the requirements and precautions the User should take when attempting to integrate a different MCU into the Zigbee Tracker v2.0.
This application note will use the ATMEGA328P, a commonly used MCU, as an example for integration. This Application Note should be read in conjunction with the Schematic and Board files for the Zigbee Tracker v2.0, which can be found in the folder provided with this Application Note (Zigbee Tracker System Files > PCB Files > Eagle > Zigbee Tracker v2.sch/Zigbee Tracker v2.brd).

2.1 Minimum Specifications

The replacement MCU should provide at minimum the following to retain most of the functionality of the Zigbee Tracker v2.0's STM32F1 series MCU:

- Power-on reset
- 5 spare GPIO (of which 2 are able to accept external interrupts)
- Hardware UART without flow control
- 100kHz capable standard I2C bus
- 3.3V (or lower) operation

Additionally, it is recommended to have the following:

- External crystal oscillator (for timing sensitive applications)
- Onboard USB
- Additional GPIO for LED status indication

It should also be noted that all GPIO (except LED indication and interrupts) must be able to hold their states in sleep mode.

2.2 Necessary Connections

The following connections to the MCU are necessary and are critical to the functionality of the Zigbee Tracker v2.0:

- Baud rate reset - Zigbee Module pin P1.7 - falling edge triggered
- AT mode set - Zigbee Module pin P1.6 - low = HEX, high = AT
- Zigbee Module reset - Zigbee Module pin RESET - active low
- IMU interrupt - BMI160 pin INT1 - active low

- Light sensor interrupt - OPT3001 pin INT - open drain

More details on pin names can be found in the folder provided with this Application Note (Zigbee Tracker System Files > PCB Files > Eagle > Zigbee Tracker v2.sch).

2.3 Example - Integrating ATMEGA328P into Zigbee Tracker v2.0

The ATMEGA328P is an 8-bit microcontroller made by Atmel. It is popular amongst hobbyists and students as there is a large community built around it. In comparison to the STM32, there are significant differences. This Section aims to replace the Zigbee Tracker v2.0's STM32F103 with an ATMEGA328P for Arduino IDE compatibility.

The differences between the ATMEGA328P and originally used STM32F103 that affect the integration into the Zigbee Tracker v2.0 are listed in the table:

	STM32F103	ATMEGA328P
Built-in USB	Present	Not present
Operating Voltage	2.0 - 3.6V	2.7 - 5.5V
GPIO Voltage Tolerance	Selected pins, V_{DD} + 4.0V	V_{DD} + 0.5V

Operating Voltage/Frequency

Even though the ATMEGA328P has a higher maximum voltage, it must be run at 3.3V (or less), due to the voltage tolerances of the other components (Zigbee module, sensors). It is highly discouraged to run the ATMEGA328P at a different voltage than the other components.

Based on Atmel's guidelines in Figure 2.3a, running the ATMEGA328P at the common 16MHz (used in Arduino) frequency requires 3.78V; the User will need to run it at a lower frequency (i.e. 8MHz).

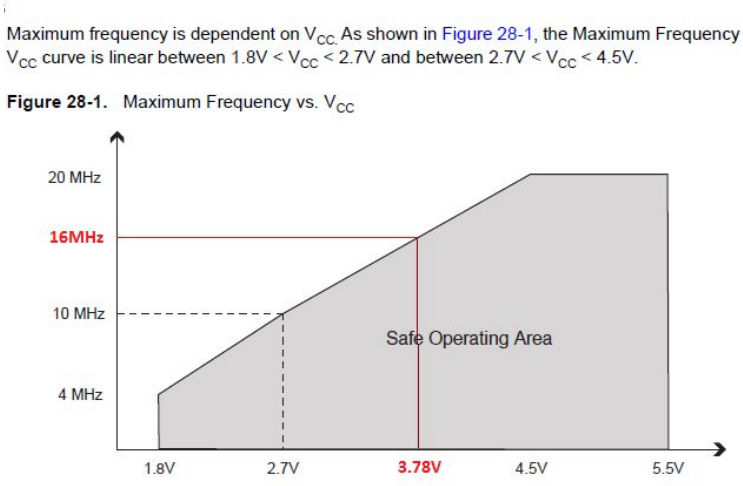

Figure 2.3a: ATMEGA328P voltage/frequency graph

GPIO Voltage Tolerance

When run at 3.3V, none of the ATMEGA328P GPIOs can be exposed to a 5V signal (i.e. to an external board or debugger) without level shifting.

Built-in USB

The STM32F103 provides built-in USB, allowing for minimal external circuitry. USB functionality is used in the case of the Zigbee Tracker v2.0 for simple programming/debugging without the need for external debuggers, and for simple serial data communication with a PC.
Depending on the requirements of the User, USB functionality may not be needed (i.e. if the User utilizes SPI programmer/AVR-ISP).

This Section will discuss the implementation of USB for ATMEGA328P. Due to the lack of built-in USB, the ATMEGA328P will need an external USB interface.
First, the standard Arduino Optiboot bootloader will need to be flashed onto the ATMEGA328P via SPI. Once flashed, the ATMEGA328P will be capable of being programmed via UART, which is easily convertible to USB (SPI to USB is harder).

Note: In order to flash the Optiboot bootloader, the ATMEGA328P SPI pins should be accessible, to use the Arduino as ISP method.

Recommended implementation (low cost) - CH340G

Figure 2.3b shows a typical circuit for the CH340G. The TXD, RXD signals are the UART connections to the ATMEGA328P, and the D+/D- pair are USB differential signals. Additionally, the ATMEGA328P RESET pin should be connected to the CH340G DTR pin through a 1uF ceramic capacitor to provide a reset pulse.
Cost per IC @ 1ku (USD): $0.304 (Q4 19)
Minimum external components: 6

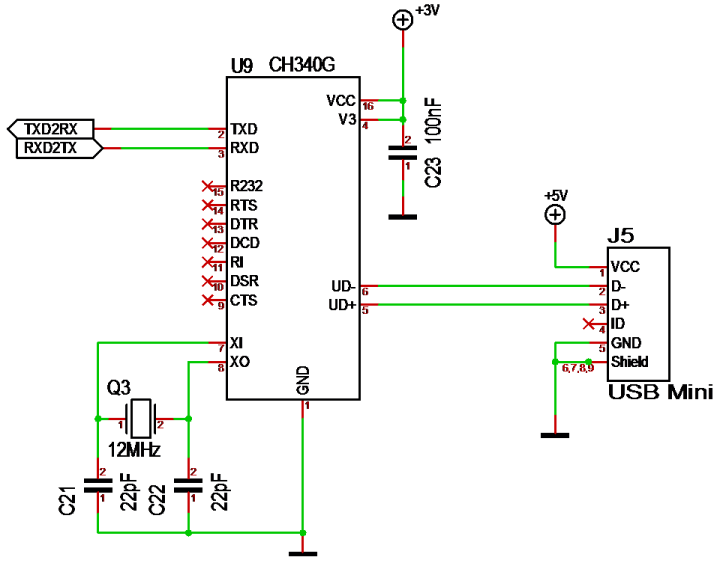

Figure 2.3b: Typical circuit for CH340G 3.3V operation

Recommended implementation (small footprint) - CP2102

Figure 2.3c shows a typical circuit for the CP2102. The TXD, RXD signals are the UART connections to the ATMEGA328P, and the D+/D- pair are USB differential signals. Additionally, the ATMEGA328P RESET pin should be connected to the CP2102 DTR pin through a 1uF ceramic capacitor to provide a reset pulse.

Cost per IC @ 1ku (USD): $1.316 (Q4 19)

Minimum external components: 3

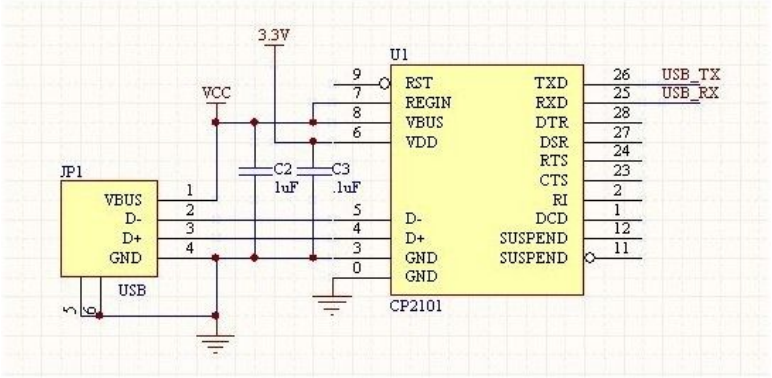

Figure 2.3c: Typical circuit for CP2102 3.3V operation