Connect to a Wi-Fi network and debug wirelessly in less than a minute

Dual Tibbo BASIC/C apps

The WM2000 is the first Tibbo device that can store **up to two** compiled Tibbo BASIC/C binaries (only one can run at any given time). All our earlier products store a single app that can take up all of the flash space not occupied by system files and TiOS. The WM2000 — which has at least four times the flash storage of any predecessor — allows you to store the second app in the space left over from the first one.

A Device Configuration Block (DCB) of the WM2000 allows you to define which of the two apps runs when the device is powered up or reboots (there is an override through the MD button/line that forces execution of the first application). A web app called BLE Terminal is provided for interacting with the DCB through the WM2000's *BLE console*. Tibbo BASIC/C applications can access the DCB through a provided API.

Simplified Wi-Fi connectivity

The WM2000 features a significantly revamped Wi-Fi API facilitating automatic association with a designated wireless network. This makes the module's Wi-Fi interface as easy to use as the Ethernet port on "wired-first" Tibbo devices. Set the target Wi-Fi network's SSID and password, enable "autoconnect," and the WM2000 will automatically associate with the network and keep associated when in range! This also enables wireless debugging of your Tibbo BASIC/C apps.

All parameters related to automatic Wi-Fi association are stored in the DCB and can be configured via the BLE console or through Tibbo BASIC/C code. A companion app that ships with the module also allows for configuration of the Wi-Fi interface.

Key Features

High-performance ARM CPU

Stores **two** Tibbo BASIC/C apps

Integrated Wi-Fi connectivity (802.11a/b/g/n)

Wireless debugging via Wi-Fi

TLS1.2 (RSA-2048)

Integrated Bluetooth Low Energy connectivity (BLE 4.2)

Supports over-the-air (OTA) updates

Two UARTs support serial, Wiegand, and clock/data streams

10 general-purpose I/O lines

Three-channel ADC

Nine PWM channels

Can drive an externally connected buzzer

Supports external matrix and binary output keypads

4MB flash for TiOS and **two** apps + 4MB for the flash disk file system

2,048-byte EEPROM

3.3V power

Onboard RTC (backup power connected externally)

Prototyping-friendly 2.54mm (100mil) pin pitch

WM2000 is Supported by Cody, Our Template Project Generator

Cody is a template project generator that produces working Tibbo BASIC code from the information you provide via a few simple-to-understand configuration screens.

This web-based app allows you to quickly create a working project that contains proper peripheral, port, and socket initialization. With Cody, you can also swiftly pour in huge chunks of your future product's functionality: persistent storage of parameters (setting), data tables, multi-level onscreen setup menus, and much much more.

No less important is that Cody generates a template project with the right structure. Software developers know that all programming platforms have (a few) right and (many) wrong ways to structure the code and get things done. Choosing the wrong path usually makes the code slow, inefficient, and unreliable. Cody prevents you from getting lost by giving you the right scaffolding for your project.

Hardware

Specifications:

- 32-bit architecture
- Powered by Tibbo OS (TiOS)
- Stores **up to two** compiled Tibbo BASIC/C binaries (apps)[1]
    - A **Device Configuration Block (DCB)**[2] defines which of the two apps normally runs on power-up
    - Forced launch of APP0 through the MD line/button
- Built-in Wi-Fi interface (802.11a/b/g/n)
    - Controlled via a simple-to-use, yet sophisticated API
    - TLS1.2 with RSA-2048 cryptosystem[3]
    - Optional "autoconnect" — automatic association with a designated Wi-Fi network as defined by the DCB[2]
    - Optional debugging of Tibbo BASIC/C applications via the Wi-Fi interface[4]
- Built-in Bluetooth Low Energy (BLE 4.2)
    - Controlled via a simple-to-use, yet sophisticated API
    - Can access the DCB via a new, integrated console[2]
- Onboard antenna (U.FL connector for an external antenna on request)
- Two high-speed serial ports (CMOS-level):
    - Baudrates of up to 921,600
    - None/even/odd/mark/space parity modes
    - 7 or 8 bits/character
    - Full-duplex mode with RTS/CTS and XON/XOFF flow control
    - Half-duplex mode with direction control
    - Encoding and decoding of Wiegand and clock/data streams
    - One of the ports can function as a serial debugging port[5]
- 10 general-purpose I/O lines
    - Two lines can work as interrupts
    - Nine lines can provide pulse-width modulation (PWM) output
    - Three lines can work as ADC inputs
- Support for externally connected matrix and binary output keypads
- RTC with dedicated backup power input
    - Only 30µA power draw on the backup power input[6]
- 58KB SRAM for Tibbo BASIC/C variables and data
- 4MB flash for code storage
    - System files and TiOS occupy a combined 2,408KB
    - 1,688KB available for storing **up to two** app binaries[1]
- **Additional** 4MB flash for the hardened fault-tolerant file system
- 2,048-byte EEPROM for data storage.
- Three onboard status LEDs
    - Green and red main status LEDs/lines

  - Yellow Wi-Fi/Bluetooth link LED/line
- Reliable power-on reset (no brown-out detection)
- Power: 150mA @ 3.3V (Wi-Fi on and scanning)
- Provisions for a deep power-down "sleep" mode
- Dimensions (L x W x H): 45.1 x 28.15 x 3.5mm
- Prototyping-friendly 2.54mm (100mil) pin pitch
- Operating temperature range: -40°C to +85°C
- Firmware and compiled Tibbo BASIC/C apps can be updated via:
  - Serial port
  - Wi-Fi interface
  - Bluetooth Low Energy (BLE) interface
- Tibbo BASIC/C applications can be debugged via Wi-Fi[4] or serial[5]
- Supplied with a companion app preloaded
  - The app allows editing of the DCB from the L.U.I.S. smartphone app (availabe on iOS and Android)
  - Users are free to modify the app for additional functionality
- CE and FCC certification pending

1. *Although two independent Tibbo BASIC/C compiled binaries (apps) can be stored in the WM2000's flash memory, only one can run at a time.*
2. *Several of the WM2000's configuration parameters are stored in the DCB, which is accessible via a new integrated console. Our BLE Terminal web app leverages the Web Bluetooth API (compatible with the Chrome, Chromium, Edge, and Opera web browsers) to connect to the WM2000's console. Configuration properties can also be read and set through Tibbo BASIC/C code.*
3. *TLS is supported on a single outgoing TCP connection.*
4. *To enable Wi-Fi debugging, you must enable autoconnect — automatic association with a designated Wi-Fi network. This can be accomplished via the integrated console, the companion app, or in code.*
5. *Only one serial port is available to Tibbo BASIC/C apps when in serial debug mode. If you need to use both serial ports, use Wi-Fi debugging instead. The debug mode can be selected via the integrated console or the companion app.*
6. *The power draw on the backup power input can vary up to ±15 percent depending on the configuration of the attached hardware.*

Programming

Create powerful, network-enabled applications in Tibbo BASIC and Tibbo C. Learn more

Platform objects:

- adc — provides access to three ADC channels.
- beep — generates buzzer patterns[1]
- bt — in charge of the BLE (Bluetooth Low-Energy) interface[1]
- button — monitors the MD (setup) line
- fd — manages the flash memory file system and direct sector access[1]
- io — handles I/O lines, ports, and interrupts
- kp — works with matrix and binary keypads
- pat — "plays" patterns on up to five LED pairs
- ppp — accesses the Internet over a serial modem (4G/LTE, etc.)
- **pwm — handles pulse-width modulation modulation channels**[1]
- romfile — facilitates access to resource files (fixed data)
- rtc — keeps track of date and time
- ser — controls serial ports (UART, Wiegand, clock/data modes)[1]

- sock — sockets (up to 32 UDP/TCP/HTTP sessions), **support for TLS**[1, 2]
- ssi — controls serial synchronous interface channels (SPI, I²C)
- stor — provides access to the EEPROM
- sys — in charge of general device functionality[1]
- wln — handles the Wi-Fi interface[1]

1. *These platform objects are either **new** or have changed significantly for the WM2000.*
2. *TLS1.2 with RSA-2048 cryptosystem, supported on a single outgoing TCP connection.*

Function Groups: String functions, trigonometric functions, date/time conversion functions, encryption/hash calculation functions, and more.

Variable Types: Byte, char, integer (word), short, dword, long, real, and string, as well as user-defined arrays and structures.

Alcom electronics