GHI Electronics, LLC
501 E. Whitcomb Ave.
Madison Heights, Michigan 48071
Phone: (248) 397-8856
Fax: (248) 397-8890
www.ghielectronics.com

G80 SoC Datasheet

Where Hardware Meets Software

1 Contents

2 Introduction

The G80 SoC is a powerful, low-cost, surface-mount LQFP100 System on Chip (SoC) running Microsoft's .NET Micro Framework. The .NET Micro Framework enables the SoC to be programmed from Microsoft Visual Studio using a USB or serial cable. Programming in a modern managed language, such as C# or Visual Basic, allows developers to accomplish more work in less time by taking advantage of the extensive built-in libraries for networking, file systems, graphical interfaces, and more.

A simple two-layer circuit board with a power source and a few connectors can utilize the G80 SoC to bring the latest technologies to any product. There are no additional licensing or other fees and all the development tools are provided freely.

Throughout this document, the G80 SoC will be referred to as the G80.

For more information and support, please see https://www.ghielectronics.com/support/netmf and the product catalog entry. For advanced electrical characteristics and details on the underlying STM32F427VGT6 processor, please consult the processor's datasheet.

2.1 Key Features

- .NET Micro Framework
- RoHS Lead Free
- 180 MHz ARM Cortex-M4 STM32F427VGT6
- 156 Kbytes available RAM
- 256 Kbytes available flash
- 78 GPIO
- 16 interrupt capable GPIO
- 2 SPI
- 1 I2C
- 4 UART
- 2 CAN
- 26 PWM
- 16 12-bit analog input
- 2 12-bit analog output
- 4-bit SD/MMC memory card interface
- Low power modes
- -40°C to +85°C operational
- LQFP100 14 x 14 mm
- RTC
- Watchdog
- Threading
- USB host
- USB client
- TCP/IP
  - Full .NET socket interface
  - Ethernet
  - PPP
- Graphics
  - Images
  - Fonts
  - Controls
- File System
  - Full .NET file interface
  - SD cards
  - USB drives
- Native extensions
  - Device register access
- Signal controls
  - Generation
  - Capture
  - Pulse measurement

2.2 Example Applications

- Vending machines
- POS Terminals
- Measurement tools and testers
- Networked sensors
- Robotics
- Central alarm system
- Smart appliances
- Industrial automation devices

3 The .NET Micro Framework

Inspired by the full .NET Framework, Microsoft developed a lightweight version called .NET Micro Framework (NETMF). NETMF focuses on the specific requirements of resource-constrained embedded systems. Development, debugging, and deployment are all conveniently performed using Microsoft's powerful Visual Studio through a standard USB or serial cable.

Programming is done in C# or Visual Basic with libraries that cover sockets, memory management with garbage collection, advanced file system support, multitasking services, and many others. In addition to supporting many standard .NET features, NETMF has additional embedded extensions supporting microcontroller specific needs such as PWM outputs and analog inputs.

3.1 GHI Electronics and NETMF

Since signing the partnership agreement with Microsoft in 2008, GHI Electronics has become the leading Microsoft partner on NETMF through its work on integrating and extending the NETMF core. GHI Electronics's NETMF products are extended with important features extending the NETMF libraries such as databases, USB Host, Wi-Fi, and native programming.

4 Pinout Table

Many signals on the G80 are multiplexed to offer multiple functions on a single pin. Developers can decide on the pin functionality to be used through the provided libraries. Any pin with no name, function, or note must be left unconnected.

1	PE2		
2	PE3	LDR0	
3	PE4	LDR1	
4	PE5		
5	PE6		
6		VBAT	
7[5]	PC13		
8[5]	PC14	RTC XTAL IN	
9[5]	PC15	RTC XTAL OUT	
10		GND	
11		3.3 V	
12		SYS XTAL IN	
13		SYS XTAL OUT	
14		RESET	
15	PC0	ADC10	
16	PC1	ADC11	
17	PC2	ADC12	SPI2 MISO
18	PC3	ADC13	SPI2 MOSI
19		3.3 V	
20[6]		GND	
21		3.3 V	
22[6]		3.3 V	
23	PA0	ADC0	COM4 TX
24	PA1	ADC1	COM4 RX
25	PA2	ADC2	PWM20
26	PA3	ADC3	PWM21
27		GND	
28		3.3 V	
29	PA4	ADC4	DAC1
30	PA5	ADC5	DAC2
31	PA6	ADC6	PWM24
32	PA7	ADC7	PWM25
33	PC4	ADC14	
34	PC5	ADC15	

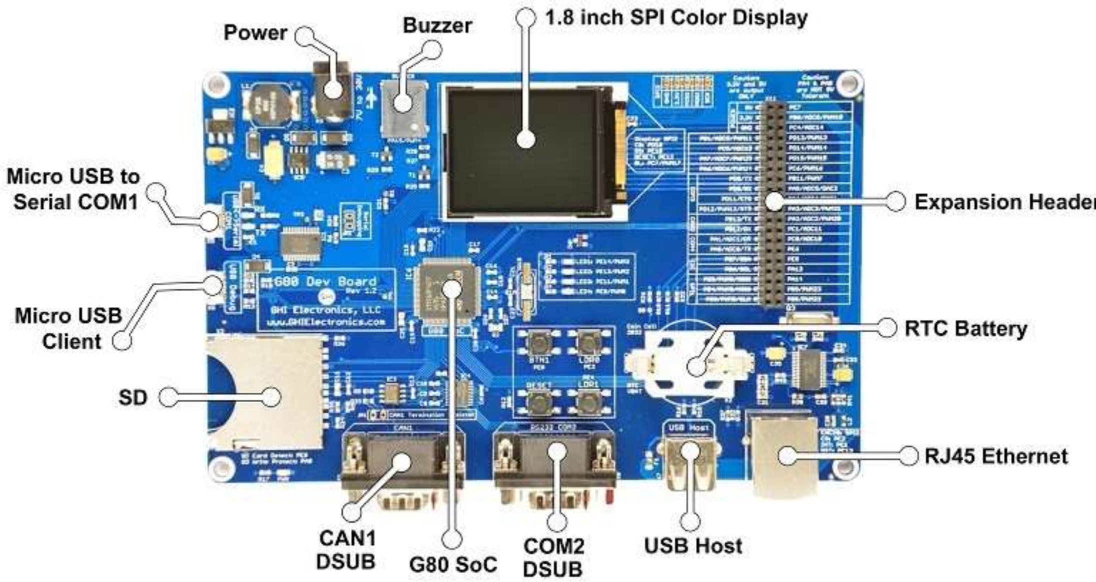
35	PB0	ADC8	PWM10
36	PB1	ADC9	PWM11
37[1]	PB2		
38	PE7		
39	PE8		
40	PE9	PWM0	
41	PE10		
42	PE11	PWM1	
43	PE12		
44	PE13	PWM2	
45	PE14	PWM3	
46	PE15	MODE	
47	PB10	PWM6	SPI2 SCK
48[2]	PB11	PWM7	
49[3]			
50		3.3 V	
51	PB12	CAN2 RD	
52	PB13	CAN2 TD	
53	PB14	USBH D-	
54	PB15	USBH D+	
55	PD8	COM3 TX	
56	PD9	COM3 RX	
57	PD10		
58	PD11	COM3 CTS	
59	PD12	COM3 RTS	PWM12
60	PD13	PWM13	
61	PD14	PWM14	
62	PD15	PWM15	
63	PC6	PWM16	
64	PC7	PWM17	
65	PC8	PWM18	SD D0
66	PC9	PWM19	SD D1
67	PA8		
68	PA9	COM1 TX	

69[2]	PA10	COM1 RX	
70	PA11	USBC D-	
71	PA12	USBC D+	
72	PA13		
73[3]			
74		GND	
75		3.3 V	
76	PA14		
77	PA15	PWM4	
78	PC10	SD D2	
79[2]	PC11	SD D3	
80	PC12	SD CLK	
81	PD0	CAN1 RD	
82	PD1	CAN1 TD	
83	PD2	SD CMD	
84	PD3	COM2 CTS	
85	PD4	COM2 RTS	
86	PD5	COM2 TX	
87	PD6	COM2 RX	
88	PD7		
89	PB3	PWM5	SPI1 SCK
90	PB4	PWM8	SPI1 MISO
91[2]	PB5	PWM9	SPI1 MOSI
92[4]	PB6	I2C SCL	
93[4]	PB7	I2C SDA	
94[1]			
95	PB8	PWM22	
96	PB9	PWM23	
97	PE0		
98	PE1		
99		GND	
100		3.3 V	

[1]Requires a 10 kΩ pull-down resistor
[2]Requires a 10 kΩ pull-up resistor
[3]Requires a 2.2 µF capacitor to GND
[4]Open drain requiring a 2.2 kΩ pull-up resistor
[5]Can only sink up to 3 mA, cannot source
[6]Used for the analog system

5 Reference Design

The G80 Dev Board is an excellent starting point and reference design for anyone interested in evaluating and developing with the G80. See the product catalog entry for more information and additional resources.



6 Device Startup

The G80 is held in reset when the reset pin is low. Releasing it will begin the system startup process.

There are three different components of the device firmware:

1. GHI Bootloader: initializes the system, updates TinyCLR when needed, and executes TinyCLR.
2. TinyCLR: loads, debugs, and executes the managed application.
3. Managed application: the program developed by the customer.

Which components get executed on startup can be control by manipulating the LDR0 pin. It is pulled high on startup. When low, the device waits in the GHI Bootloader. Otherwise, the managed application is executed. LDR1 is reserved for future use.

Additionally, the communications interface between the host PC and the G80 is selected on startup through the MODE pin, which is pulled high on startup. The USB interface is selected when MODE is high and COM1 is selected when MODE is low.

The above discussed functions of LDR0, LDR1, and MODE are only during startup. After startup, they return to the default GPIO state and are available to use as GPIO in the user application.

7 Libraries

Similar to the full .NET Framework, NETMF includes many built in libraries to help in modern application development with additional libraries to support embedded systems.

Please see https://www.ghielectronics.com/support/netmf for more information.

7.1 General Purpose Input and Output (GPIO)

GPIOs can read and write logical high and low signals. Keep the following in mind:

- They default to inputs with internal weak pull-up resistors
- They operate on 3.3 V logic levels.
- They are 5 V tolerant when not in analog or crystal mode, though analog output is never tolerant.
- They have controllable pull up and pull down resistors.
- Not all pins are interrupt capable at the same time. See design considerations.
- Most pins can source or sink up to 8 mA (see the processor's documentation for advanced information).

7.2 Analog Input

Analog inputs can read voltages from 0 V to 3.3 V with 12-bit resolution. The built in analog circuitry uses the source voltage as a reference which can cause some noise on the analog signal. High accuracy ADCs with a dedicated reference can be added externally.

7.3 Analog Output

Analog outputs can vary their voltage from 0 V to 3.3 V with 12-bit resolution. The output voltage is meant to be used as a signal and not a driver. An op-amp or similar circuit can be used to amplify the current.

7.4 Pulse Width Modulation (PWM)

PWM is used to create a waveform with a specified frequency and duty cycle. It uses built-in hardware so no processing resources are needed to keep it running. Frequencies can range from 1 Hz to 90 MHz.

Some PWM channels share the same source clock internally. Changing the frequency on a channel will affect other channels; however, they can have a separate duty cycle.

Channel	Timer
0 to 3	1
4 to 7	2
8 to 11	3
12 to 15	4
16 to 19	8
20 and 21	9

7.5 Signal Generator

Signal Generator is used to generate a waveform on any GPIO with varying frequency and duty cycle. The feature is software driven and can generate frequencies up to 250 kHz ±10%. More processing time is required for higher frequencies.

7.6 Signal Capture

Signal Capture monitors any GPIO and records the time from the last change. This feature is software driven and can measure frequencies up to 400 kHz ±10%. Lower frequencies have higher accuracy.

7.7 Pulse Feedback

Pulse Feedback is used for sensing capacitance on any input pin and measuring pulses from ultrasonic distance and other sensors. When used for sensing capacitance, a 100 pF capacitor and 1 MΩ resistor between the pad and ground are recommended.

7.8 Universal Asynchronous Receiver Transmitter (UART)

UART is a common, full duplex, communications interface. Baud rates from 2,400 to 921,600 are supported. Handshaking is supported on COM2 and COM3 only. Data bits of 8 and 9 are supported. Stop bits of 1 and 2 are supported. Even and odd parities are supported.

7.9 Serial Peripheral Interface (SPI)

SPI is a common three or four wire serial interface. The G80 can act as a SPI bus master only. The maximum supported clock is 45 MHz and all four SPI modes are supported. The SPI bus is designed to interface with multiple SPI slave devices. The active slave is selected by asserting the chip select line on the slave device.

7.10 Inter-Integrated Circuit (I2C)

I2C is a two-wire addressable serial interface. The G80 can act as an I2C bus master only with 7-bit slave addresses. It can connect to one or more slave devices over the same connection with a maximum clock of 400 kHz. The I2C bus interface requires pull up resistors to be added on both the SCL and SDA pins, usually 2.2 kΩ.

It is possible to simulate an independent I2C bus on any two GPIO pins with the appropriate resistors though the software I2C class, but performance will be lower.

7.11 Controller Area Network (CAN)

CAN is a common interface in industrial control and the automotive industry. CAN on the G80 is compliant with the CAN 2.0B specifications. Bitrates up to 1 Mbit/s are supported. For systems with higher traffic, different message filter options are available.

7.12 1-Wire

Through 1-Wire, a master can communicate with multiple 1-Wire slaves using any GPIO.

7.13 Graphics

The G80 supports SPI displays and drawing though the NETMF bitmap object. TrueType font files can be used once converted to the TinyFont format used by NETMF.

7.14 Touch Screen

Capacitive touch displays can be used through the I2C interface.

7.15 USB Host

USB host allows the use of USB mass storage devices, joysticks, keyboards, and mice. Additionally, for USB devices that do not have a standard class included, low level USB access is provided for bulk transfers.

7.16 USB Client

The USB client interface is typically used as the G80 debug interface and for application deployment through Visual Studio. However, it is controllable and may be used to simulate other USB devices such as mice, keyboards, and Communications Device Class (CDC) interfaces using low level access instead of the debug interface.

7.17 File System

The G80 supports accessing files on SD cards and USB memory devices formatted as FAT16 or FAT32. SD cards use the true 4-bit interface. MMC/SD/SDHC/SDXC cards in full, mini, and micro formats and any USB device with mass storage class are supported. Access speeds are dependent on many different factors and can be up to 500 Kbyte/s.

7.18 Networking

The G80 supports Ethernet and PPP through the built in LwIP stack. The full stack includes TCP, UDP, DHCP, DNS, HTTP, FTP, and others.

7.18.1 Ethernet

Ethernet support is available using the built-in NETMF TCP/IP stack through an external ENC28J60 SPI Ethernet chip.

7.18.2 Wi-Fi

There is no internal support for Wi-Fi. However, any Wi-Fi module with a built-in TCP/IP stack can be used.

7.18.3 Point to Point

The Point to Point (PPP) protocol is often used for devices needing to connect to mobile networks. While typical embedded devices use the mobile modem's built-in and very limited TCP/IP stack, systems using the G80 can use these modems with the internal NETMF TCP/IP.

7.19 Configuration

Access to the configuration sector of the device is provided for storage of small, infrequently changing, entries. The data will be lost if the configuration is reflashed. Space is limited and varies based on other information stored in the configuration.

7.20 Real Time Clock

The real time clock (RTC) is used to keep time while the processor is off, drawing its power from a backup battery or super capacitor providing 1.65 V to 3.6 V. An appropriate 32,768 Hz crystal and its associated circuitry must be connected to the G80 for the RTC to function.

7.21 Watchdog

Watchdog is used to reset the system if it enters an erroneous state. The G80 supports timeouts between 1 ms and 32,768 ms. Watchdog support is included through the GHI Electronics libraries replacing the built in NETMF version.

7.22 Power Control

The G80 supports entering sleep, deep sleep, and off modes in order to reduce power usage. It can consume as little as 40 mA in sleep, 12 mA in deep sleep, and 8 mA in off. It may be woken from an RTC alarm or a GPIO interrupt. Sleep pauses execution of the program. Deep sleep pauses execution of the program and shuts down many internal functions. Off shuts down all internal functions and can only be woken by the RTC alarm or a system reset. The system will be automatically reset when exiting off mode.

7.23 Direct Memory Access

Low level device registers and memory can be accessed to further configure the G80's underlying processor. Not all functionality of the processor is available as some functions may be used or configured internally for use in NETMF.

7.24 Battery RAM

Battery-backed RAM is provided as part of the internal RTC. This memory retains its contents when the power is lost as long as there is a backup battery. There are 4 Kbytes of battery backed RAM available. Consult the processor's documentation for details on use.

8 Design Considerations

8.1 Required Pins

Exposing the following pins is required in every design to enable device programming, updates, and recovery:

- LDR0
- LDR1
- Desired debug interface(s)
- MODE if required to select a debug interface

8.2 Power Supply

A typical clean power source, suited for digital circuitry, is needed to power the G80. Voltages should be within at least 10% of the needed voltage. Decoupling capacitors of 0.1 µF are needed near every power pin. Additionally, a large capacitor, typically 47 µF, should be near the G80 if the power supply is more than few inches away.

8.3 Crystals

The G80 requires an external 12 MHz crystal and associated circuitry to function. For the RTC to function, a 32,768 Hz crystal and circuitry are required. Please see the processor's documentation for advanced information.

8.4 Interrupt Pins

Interrupts are only available on 16 pins at any given time. Of those 16 pins, the pin number must be unique. For example: PA1 and PB1 cannot both be used as interrupts at the same time, but PA1 and PB2 can.

8.5 Reset

The reset pin is not pulled in any direction. Designs must be sure to use an appropriate pull-up resistor.

8.6 Direct Memory Access

Most of the core processor's resources are used by NETMF. Some resources are permanently used, like the main system timer while others are used when specific features, like the timers for PWM, are enabled. Used resources can change from one firmware version to another so care must be taken when using these resources through direct memory access methods.

When absolutely required, applications can use resources in conjunction with NETMF. For example, creating a special baud rate, utilizing the timer capture feature, and making use of many other features supported by the processor. Please contact GHI Electronics's consulting services to determine exactly what resources are available and if the G80 can fulfill the specific requirements.

9 Legal Notice

9.1 Licensing

The G80 SoC, with all its built-in software components, is licensed for commercial and non-commercial use. No additional fee or licensing is required. Software, firmware, and libraries provided for the G80 SoC are licensed to be used on the G80 SoC only.

9.2 Trademarks

G80 is a trademark of GHI Electronics, LLC.

.NET Micro Framework and Visual Studio are registered or unregistered trademarks of Microsoft Corporation.

Other registered or unregistered trademarks are owned by their respective companies.

9.3 Disclaimer

IN NO EVENT SHALL GHI ELECTRONICS, LLC BE LIABLE FOR ANY DIRECT, INDIRECT, INCIDENTAL, SPECIAL, EXEMPLARY, OR CONSEQUENTIAL DAMAGES (INCLUDING, BUT NOT LIMITED TO, PROCUREMENT OF SUBSTITUTE GOODS OR SERVICES; LOSS OF USE, DATA, OR PROFITS; OR BUSINESS INTERRUPTION) HOWEVER CAUSED AND ON ANY THEORY OF LIABILITY, WHETHER IN CONTRACT, STRICT LIABILITY, OR TORT (INCLUDING NEGLIGENCE OR OTHERWISE) ARISING IN ANY WAY OUT OF THE USE OF THIS PRODUCT, EVEN IF ADVISED OF THE POSSIBILITY OF SUCH DAMAGE. GHI ELECTRONICS, LLC LINE OF PRODUCTS ARE NOT DESIGNED FOR LIFE SUPPORT APPLICATIONS. SPECIFICATIONS AND AVAILABILITY ARE SUBJECT TO CHANGE WITHOUT ANY NOTICE.

10 Revision History

Revision	Date	Change
1.0	2015-11-12	Initial release.