STM32G071RB IoT Project User Manual

Embedded Systems Project

July 6, 2025

Abstract

This user manual provides comprehensive documentation for the STM32G071RB IoT prototype system featuring LoRa wireless communication and BME680 environmental sensor integration. The system enables wireless sensor data transmission, environmental monitoring, and remote command control through dual UART interfaces. This manual covers hardware setup, software configuration, command interface usage, troubleshooting, and development guidelines.

Contents

1 System Overview

1.1 Project Description

The STM32G071RB IoT prototype is an embedded system designed for wireless sensor data transmission and environmental monitoring. The system integrates a BME680 environmental sensor for temperature, pressure, and humidity measurements, and an SX126x LoRa module for long-range wireless communication.

1.2 Key Features

- **Environmental Sensing**: BME680 sensor for temperature, pressure, and humidity
- **Wireless Communication**: LoRa (SX126x) for long-range data transmission
- **Dual UART Interface**: USART2 and USART4 for command and debugging
- **I2C Communication**: For BME680 sensor interface
- **SPI Communication**: For LoRa module interface
- **Real-time Monitoring**: Continuous sensor data acquisition and transmission
- **Command Interface**: Interactive command system for system control

1.3 Technical Specifications

Parameter	**Specification**
Microcontroller	STM32G071RBT6
System Clock	16 MHz
Operating Voltage	3.3V
LoRa Frequency	868 MHz
LoRa Spreading Factor	SF7
LoRa Bandwidth	125 kHz
UART Baud Rate	115200 bps
I2C Speed	100 kHz
SPI Speed	8 MHz

Table 1: System Technical Specifications

2 Hardware Setup

2.1 Required Components

- STM32G071RB Nucleo-64 board
- BME680 environmental sensor module
- SX126x LoRa module (e.g., SX1262, SX1268)
- USB-to-UART converter (for USART4)
- Breadboard and jumper wires
- 4.7kOhm pull-up resistors (for I2C)
- Power supply (3.3V)

2.2 Pin Connections

2.2.1 STM32G071RB Pinout

Pin	Function	Connection
PA2	USART2 TX	USB-to-UART RX
PA3	USART2 RX	USB-to-UART TX
PA0	USART4 TX	USB-to-UART RX
PA1	USART4 RX	USB-to-UART TX
PA9	I2C1 SCL	BME680 SCL
PA10	I2C1 SDA	BME680 SDA
PA5	SPI1 SCK	LoRa SCK
PA6	SPI1 MISO	LoRa MISO
PA7	SPI1 MOSI	LoRa MOSI
PA4	SPI1 NSS	LoRa NSS
PC0	GPIO	LoRa RESET
PA8	GPIO	LoRa DIO1

Table 2: STM32G071RB Pin Connections

2.2.2 BME680 Sensor Connections

BME680 Pin	STM32 Pin	Description
VCC	3.3V	Power supply
GND	GND	Ground
SCL	PA9	I2C clock line
SDA	PA10	I2C data line

Table 3: BME680 Sensor Connections

2.2.3 LoRa Module Connections

LoRa Pin	STM32 Pin	Description
VCC	3.3V	Power supply
GND	GND	Ground
SCK	PA5	SPI clock
MISO	PA6	SPI data in
MOSI	PA7	SPI data out
NSS	PA4	SPI chip select
RESET	PC0	Module reset
DIO1	PA8	Interrupt line

Table 4: LoRa Module Connections

3 Software Architecture

3.1 Project Structure

The project follows a modular architecture with the following components:

- **main.c**: System initialization and main loop
- **bme680_interface.c**: BME680 sensor driver and interface
- **lora_interface.c**: LoRa communication interface
- **command_interface.c**: Command processing and user interface
- **sx126x.c**: SX126x LoRa driver (external library)

3.2 System Initialization Flow

1. Hardware initialization (GPIO, UART, I2C, SPI)
2. BME680 sensor detection and initialization
3. LoRa module detection and configuration
4. Command interface startup
5. Main loop execution

3.3 Communication Protocols

3.3.1 I2C Protocol (BME680)

- **Clock Speed**: 100 kHz
- **Device Address**: 0x76 (default) or 0x77
- **Data Format**: 8-bit data, 7-bit address
- **Pull-up Resistors**: 4.7kOhm recommended

3.3.2 SPI Protocol (LoRa)

- **Clock Speed**: 8 MHz
- **Mode**: SPI Mode 0 (CPOL=0, CPHA=0)
- **Data Order**: MSB first
- **Chip Select**: Active low

3.3.3 UART Protocol

- **Baud Rate**: 115200 bps
- **Data Bits**: 8
- **Parity**: None
- **Stop Bits**: 1
- **Flow Control**: None

4 Command Interface

4.1 Getting Started

1. Connect the hardware according to the wiring diagram
2. Power on the STM32 board
3. Open a terminal application (PuTTY, Tera Term, etc.)
4. Configure the terminal:
   - Baud Rate: 115200
   - Data Bits: 8
   - Parity: None
   - Stop Bits: 1
   - Flow Control: None
5. Connect to the appropriate COM port
6. Type `start` to begin the command interface

4.2 Available Commands

4.2.1 Sensor Commands

Command	Alias	Description
`read temperature`	`rt`	Read temperature from BME680
`read pressure`	`rp`	Read pressure from BME680
`read humidity`	`rh`	Read humidity from BME680
`test sensor`	`ts`	Test BME680 sensor functionality
`raw registers`	`rr`	Read raw BME680 registers
`raw adc`	`ra`	Read raw BME680 ADC values
`calib data`	`cd`	Check BME680 calibration data
`scan i2c`	`si`	Scan I2C bus for devices

Table 5: Sensor Commands

4.2.2 LoRa Commands

Command	Alias	Description
`lora broadcast`	`lb`	Broadcast sensor data via LoRa
`lora config`	`lc`	Show LoRa configuration
`lora test`	`lt`	Test LoRa transmission
`lora scan`	`ls`	Scan for LoRa signals (5s)
`lora monitor`	`lm`	Start continuous monitoring
`lora stop`	`lst`	Stop LoRa monitoring
`lora rssi`	`lr`	Get current RSSI

Table 6: LoRa Commands

Command	Description
`help`	Show help menu
`start`	Start the command interface

Table 7: System Commands

4.2.3 System Commands

4.2.4 Math Operations

Command	Description
`sum <num1> <num2>`	Add two numbers
`sub <num1> <num2>`	Subtract num2 from num1
`mul <num1> <num2>`	Multiply two numbers
`div <num1> <num2>`	Divide num1 by num2

Table 8: Math Operations

4.3 Command Examples

4.3.1 Sensor Data Reading

```
> start
System started! Type 'help' for available commands.
> rt
Temperature: 23.45 C
> rp
Pressure: 1013.25 hPa
> rh
Humidity: 45.67%
> ts
Sensor Test Results:
- Temperature: 23.45 C
- Pressure: 1013.25 hPa
- Humidity: 45.67%
- Status: OK
```

Listing 1: Reading Sensor Data

4.3.2 LoRa Communication

```
> lb
Broadcasting sensor data via LoRa...
Message sent: {"temp":23.45,"press":1013.25,"hum":45.67,"node":"STM32"}
> lc
LoRa Configuration:
- Frequency: 868000000 Hz
- Spreading Factor: SF7
- Bandwidth: 125 kHz
- Coding Rate: 4/5
- TX Power: 14 dBm
- Sync Word: 0x12
> ls
Scanning for LoRa signals...
RSSI: -85 dBm
Signal detected at 868.0 MHz
```

Listing 2: LoRa Communication

5 Troubleshooting

5.1 Common Issues and Solutions

5.1.1 BME680 Sensor Issues

Problem	**Cause**	**Solution**
Sensor not detected	Wrong I2C address	Check SDO pin connection
Communication errors	Missing pull-up resistors	Add 4.7kOhm resistors to SCL/SDA
Invalid readings	Power supply issues	Ensure 3.3V stable supply
I2C bus errors	Wiring issues	Verify SCL/PA9 and SDA/PA10 connections

Table 9: BME680 Troubleshooting

5.1.2 LoRa Module Issues

Problem	**Cause**	**Solution**
Module not detected	SPI wiring issues	Check SCK, MISO, MOSI, NSS connections
Transmission failures	Antenna not connected	Connect proper LoRa antenna
Low signal strength	Wrong frequency	Verify 868 MHz configuration
Communication errors	SPI speed too high	Reduce SPI clock frequency

Table 10: LoRa Troubleshooting

5.1.3 UART Communication Issues

Problem	**Cause**	**Solution**
No communication	Wrong COM port	Check Device Manager for correct port
Garbled text	Wrong baud rate	Set to 115200 bps
Missing characters	Flow control enabled	Disable hardware flow control
Connection drops	Driver issues	Update USB-to-UART driver

Table 11: UART Troubleshooting

5.2 Diagnostic Commands

Use these commands to diagnose system issues:

- `scan i2c`: Check I2C bus for connected devices
- `raw registers`: Read raw BME680 registers for debugging
- `lora config`: Verify LoRa module configuration
- `lora rssi`: Check signal strength

5.3 System Status Messages

The system provides status messages during initialization:

```
========================================
IoT Prototype System - STM32G071RB
========================================
System Clock: 16 MHz
I2C1 Configuration: PA9 (SCL), PA10 (SDA)
USART2: PA2 (TX), PA3 (RX) - 115200 baud
USART4: PA0 (TX), PA1 (RX) - 115200 baud
SPI1: PA5 (SCK), PA6 (MISO), PA7 (MOSI)
LoRa: PA4 (NSS), PC0 (RESET)
LED Status: PA5
========================================
```

Listing 3: System Initialization Messages

6 Development Guidelines

6.1 Adding New Commands

To add new commands to the system:

1. Add command handler in `command_interface.c`
2. Update help menu with new command description
3. Implement command functionality
4. Test with both USART2 and USART4 interfaces

6.2 Modifying Sensor Configuration

To modify BME680 sensor settings:

```c
struct bme68x_conf conf;
conf.os_hum = BME68X_OS_2X;     // Humidity oversampling
conf.os_pres = BME68X_OS_4X;   // Pressure oversampling
conf.os_temp = BME68X_OS_2X;   // Temperature oversampling
conf.filter = BME68X_FILTER_SIZE_3; // IIR filter
conf.odr = BME68X_ODR_1000_MS; // Output data rate
```

Listing 4: BME680 Configuration Example

6.3 Modifying LoRa Parameters

To modify LoRa communication parameters:

```c
static sx126x_mod_params_lora_t lora_mod_params = {
    .sf = SX126X_LORA_SF8,        // Spreading factor
    .bw = SX126X_LORA_BW_250,     // Bandwidth
    .cr = SX126X_LORA_CR_4_7,     // Coding rate
    .ldro = 1                     // Low data rate optimization
};
```

Listing 5: LoRa Configuration Example

7 Performance Characteristics

7.1 Sensor Performance

Parameter	Range	Accuracy
Temperature	-40C to +85C	+/-0.5C
Pressure	300 hPa to 1100 hPa	+/-1 hPa
Humidity	0% to 100%	+/-3%

Table 12: BME680 Performance Specifications

7.2 LoRa Performance

Parameter	Value	Description
Frequency	868 MHz	Operating frequency
Range	Up to 15 km	Line of sight
Data Rate	5.47 kbps	With SF7, 125 kHz BW
TX Power	14 dBm	Maximum output power

Table 13: LoRa Performance Specifications

7.3 System Performance

- **Command Response Time**: < 100 ms
- **Sensor Read Time**: < 50 ms
- **LoRa Transmission Time**: < 200 ms
- **Power Consumption**: 50 mA (active mode)

8 Advanced Features

8.1 Continuous Monitoring

The system supports continuous LoRa monitoring for detecting signals from other devices:

```
> lm
Starting continuous LoRa monitoring...
Monitoring for signals...
RSSI: -87 dBm
Packet received: {"temp":24.1,"press":1012.8,"hum":48.2,"node":"Node2"}
RSSI: -92 dBm
Signal detected (no packet)
> lst
Monitoring stopped.
```

Listing 6: Continuous Monitoring

8.2 Sensor Data Broadcasting

Automatic broadcasting of sensor data in JSON format:

```
> lb
Broadcasting sensor data via LoRa...
Message sent: {"temp":23.45,"press":1013.25,"hum":45.67,"node":"STM32"}
```

Listing 7: Sensor Data Broadcast

8.3 I2C Bus Scanning

Automatic detection of I2C devices:

```
> si
Scanning I2C bus for devices...
Device found at address: 0x76
Total devices found: 1
```

Listing 8: I2C Bus Scan

9 Maintenance and Support

9.1 Regular Maintenance

- Check physical connections monthly
- Verify sensor calibration annually
- Update firmware as needed
- Monitor system logs for errors

9.2 Calibration

The BME680 sensor includes factory calibration data. For high-precision applications:

- Use reference instruments for comparison
- Apply offset corrections in software
- Consider environmental factors
- Document calibration procedures

9.3 Technical Support

For technical support and questions:

- Check this user manual first
- Review troubleshooting section
- Verify hardware connections
- Test with known good components

10 Appendices

10.1 Appendix A: Pin Definitions

```
// UART Pins
#define UART2_TX_PIN GPIO_PIN_2
#define UART2_TX_PORT GPIOA
#define UART2_RX_PIN GPIO_PIN_3
#define UART2_RX_PORT GPIOA

#define UART4_TX_PIN GPIO_PIN_0
#define UART4_TX_PORT GPIOA
#define UART4_RX_PIN GPIO_PIN_1
#define UART4_RX_PORT GPIOA

// I2C Pins
#define I2C1_SCL_PIN GPIO_PIN_9
#define I2C1_SCL_PORT GPIOA
#define I2C1_SDA_PIN GPIO_PIN_10
#define I2C1_SDA_PORT GPIOA

// SPI Pins
#define SPI1_SCK_PIN GPIO_PIN_5
#define SPI1_SCK_PORT GPIOA
#define SPI1_MISO_PIN GPIO_PIN_6
#define SPI1_MISO_PORT GPIOA
#define SPI1_MOSI_PIN GPIO_PIN_7
#define SPI1_MOSI_PORT GPIOA
#define SPI1_NSS_PIN GPIO_PIN_4
#define SPI1_NSS_PORT GPIOA

// LoRa Control Pins
#define LORA_RESET_PIN GPIO_PIN_0
#define LORA_RESET_PORT GPIOC
#define LORA_DIO1_PIN GPIO_PIN_8
#define LORA_DIO1_PORT GPIOA
```

Listing 9: Pin Definitions in main.h

10.2 Appendix B: Configuration Constants

```
// LoRa Configuration
#define LORA_FREQUENCY_HZ        868000000
#define LORA_TX_POWER_DBM        14
#define LORA_SYNC_WORD           0x12
#define LORA_PAYLOAD_LENGTH      64

// BME680 Configuration
#define BME68X_I2C_ADDR_LOW      0x76
#define BME68X_I2C_ADDR_HIGH     0x77
#define BME68X_CHIP_ID           0x61

// System Configuration
#define CMD_BUFFER_SIZE          128
#define SYSTEM_CLOCK_HZ          16000000
```

Listing 10: Configuration Constants

Error Code	Description	Action
BME68X_OK	Operation successful	None
BME68X_E_COM_FAIL	Communication failure	Check I2C connections
BME68X_E_DEV_NOT_FOUND	Device not found	Check sensor power and address
SX126X_STATUS_OK	LoRa operation successful	None
SX126X_STATUS_ERROR	LoRa operation failed	Check SPI connections
HAL_OK	HAL operation successful	None
HAL_ERROR	HAL operation failed	Check hardware connections

Table 14: Error Codes and Descriptions

10.3 Appendix C: Error Codes

10.4 Appendix D: Command Reference

Command	Alias	Description
`start`	-	Start command interface
`help`	-	Show help menu
`read temperature`	`rt`	Read temperature (C)
`read pressure`	`rp`	Read pressure (hPa)
`read humidity`	`rh`	Read humidity (%)
`test sensor`	`ts`	Test BME680 sensor
`raw registers`	`rr`	Read raw registers
`raw adc`	`ra`	Read raw ADC values
`calib data`	`cd`	Check calibration data
`scan i2c`	`si`	Scan I2C bus
`lora broadcast`	`lb`	Broadcast sensor data
`lora config`	`lc`	Show LoRa config
`lora test`	`lt`	Test LoRa transmission
`lora scan`	`ls`	Scan for signals (5s)
`lora monitor`	`lm`	Start monitoring
`lora stop`	`lst`	Stop monitoring
`lora rssi`	`lr`	Get RSSI
`sum <n1> <n2>`	-	Add numbers
`sub <n1> <n2>`	-	Subtract numbers
`mul <n1> <n2>`	-	Multiply numbers
`div <n1> <n2>`	-	Divide numbers