

RIM51

The RIM51 is a complete embedded controller in a 40-pin DIP format.

- A High-speed 8052 microcontroller
- Up to 128K of FLASH code memory
- Up to 128K of RAM
- An RS-232 serial port
- Power-on reset circuitry
- 12 bits of 8051-type input/output ports
- 20 bits of 3.3 Volt bi-directional input/output ports (5-Volt tolerant)

Simplifies Applications, Reduces Cost

The module significantly reduces the applications hardware. Since all the necessary controller functions are encapsulated in the module, the typical applications hardware consists of a low-cost board with customized sensors and actuators with a 40-pin DIP socket to accommodate the RIM51.

Programming

The RIM51 module is programmed through its serial port. It requires only four lines (+5Volts, Ground, TxD, RxD) to communicate with a PC. The 8051 code may be developed using Rigel's IDE Reads51, or any other third party compiler/assembler. The generated HEX code is downloaded to the RIM51 by the standard Reads51 download command. A separate programming utility is also available for OEM applications.

RIM51 Pins

Pin Number	Function	Function	Pin Number
1	P1.0	VCC	40
2	P1.1	PA.0	39
3	P1.2	PA.1	38
4	P1.3	PA.2	37
5	P1.4	PA.3	36
6	P1.5	PA.4	35
7	P1.6	PA.5	34
8	P1.7	PA.6	33
9	RESET#	PA.7	32
10	PC.0	PRG#	31
11	PC.1	A16#	30
12	P3.2	A17#	29
13	P3.3	PB.7	28
14	P3.4	PB.6	27
15	P3.5	PB.5	26
16	PC.2	PB.4	25
17	PC.3	PB.3	24
18	RXD	PB.2	23
19	TXD	PB.1	22
20	GND	PB.0	21

Pin Layout of the RIM51

Power

VCC (+5 Volts) and GND.

Port 1

The eight pins P1.0 to P1.7 are 8051-type bi-directional ports. They are connected to the microcontroller's P1 port pins. P1 bits also implement the standard 8052 alternate functions.

Port 3

The four pins P3.2 to P3.5 are 8051-type bi-directional ports. They are connected to the microcontroller's P3 port pins. P3 bits also implement the standard 8052 alternate functions.

Port A

The eight pins PA.0 to PA.7 are 3.3 Volt bi-directional ports. They are 5 Volt tolerant. They are memory mapped to the microcontroller's external data memory.

Port B

The eight pins PB.0 to PB.7 are 3.3 Volt bi-directional ports. They are 5 Volt tolerant. They are memory mapped to the microcontroller's external data memory.

Port C

The four pins PC.0 to PC.4 are 3.3 Volt bi-directional ports. They are 5 Volt tolerant. They are memory mapped to the microcontroller's external data memory.

RESET#

This is an active-low input pin with an internal pull-up resistor. If left floating, the RIM51 resets upon power up. Connect RESET# to ground to induce an external reset.

PRG#

This is an active-low input pin with an internal pull-up resistor. If left floating, the RIM51 runs the code programmed into its FLASH memory. Grounding PRG# places the RIM51 in its program mode. The internal FLASH may be programmed through the serial port in this mode.

A16# and A17#

These are active-low input pins with internal pull-up resistors. They are the inverted address lines to be used when the code memory or the data memory exceeds 64K. They are not used if both the code and the data memory are 64K or smaller. They may be left floating if only the first 64K of external code and external data memory are to be used. Typically, these lines are connected to port pins to perform memory bank switching.

Port Configuration

Three of the ports PA, PB, and PC are memory-mapped bi-directional input/output ports. Use the MOVX command to read from and write to the ports.

Address	Port
0xFFFC	PA
0xFFFD	PB
0xFFFE	PC
0xFFFF	Configuration

The direction of the ports PA, PB, and PC are determined by the configuration byte written to address 0xFFFF.

Only the least significant three bits of the configuration byte are used. These are the direction bits DA, DB, and DC, corresponding to ports PA, PB, and PC. The remaining bits are reserved for future use.

Port configuration byte at address 0xFFFF.
(X: not used, reserved)

bit 7	bit 6	bit 5	bit 4	bit 3	bit 2	bit 1	bit 0
X	X	X	X	X	DC	DB	DA

Setting the direction bit configures the associated port to be an input port. Clearing the direction bit makes the associated port an output port. All ports are configured as input ports upon reset. Refer to the various example programs given in the Reads51 IDE for an illustration of configuring and using the RIM51 ports.

The following assembly code segments illustrate the use of the ports memory-mapped ports. First, the ports addresses are defined.

```
#define RIM_PORTA 0xFFFC
#define RIM_PORTB 0xFFFD
#define RIM_PORTC 0xFFFE
#define RIM_DIR 0xFFFF
```

Let Port A and Port B be configured as output ports, and Port C as an input port.

```
mov  dptr, #PORT_DIR
mov  a, #0xFC
movx @dptr, a
```

Let Port A output the value #0xAA, and Port B the value #0x55.

```
mov  dptr, #PORT_A
mov  a, #0xAA
movx @dptr, a

mov  dptr, #PORT_B
mov  a, #0x55
movx @dptr, a
```

The following code reads the value of the input port (Port C) into the accumulator.

```
mov  dptr, #PORT_C
movx a, @dptr
```

The following accomplishes the same steps in C.

```
/* define the ports */
unsigned char *pucPortA = RIM_PORTA,
              *pucPortB = RIM_PORTB,
              *pucPortC = RIM_PORTC;
              *pucPortDir = RIM_DIR;

/* set PortA to 0xAA, and PortB to 0x55 */
*pucPortA = 0xAA;
*pucPortB = 0x55;

/* read the value of PortC into the variable nX */
nX = *pucPortC;
```