

1978



**MOTOROLA**  
Semiconductor Products Inc.

**AN-782**  
Application Note

## INTERFACING AND CONTROLLING DIGITAL TEMPERATURE DATA USING THE MC6800

The MC6821 and the MC6800 coupled with a suitable digital temperature device make a valuable tool for maintaining a stable temperature in various control applications. Upper and lower temperature bounds may be set within the software providing a variable temperature window. The microprocessor can check the temperature preset by boundaries and send external signals to regulate the thermionic device. An overall system block diagram is shown in Figure 1.

Eight bits of temperature data are handwired to the MC6821 PIA. The MC6821 provides the universal means of interfacing peripheral equipment to the MC6800 MPU through two 8-bit bidirectional lines. Normally no external logic is required for interfacing to most peripheral devices.

The MC6821 is programmed by the MC6800 MPU. In this system PIA Port B was used which consists of eight lines which may be programmed as an input or output depending on how the PIA is programmed. The MC6821 is internally addressed in order to configure the data and control lines. Table 1 shows the internal addressing for the MC6821.

To set the direction of the data lines the Data Direction Register must be accessed by writing a "0" into bit 2 of the Control Register. This selects the Data Direction Register and now the corresponding address for this register (see Table 1) may be written to configure the individual lines as inputs or outputs. A Data Direction Register bit set at "0" makes the corresponding line an input and a "1" makes the corresponding line an output.

In order to access the Peripheral Register it is necessary to write a "1" into bit 2 of the Control Register. This selects the Peripheral Register which means the lines set as

TABLE 1 — INTERNAL ADDRESSING

RS1	RS0	Control Register Bit		Location Selected
		CRA-2	CRB-2	
0	0	1	X	Peripheral Register A
0	0	0	X	Data Direction Register A
0	1	X	X	Control Register A
1	0	X	1	Peripheral Register B
1	0	X	0	Data Direction Register B
1	1	X	X	Control Register B

X = Don't Care

outputs may be written into the lines set as inputs may be read from.

For example, assume the PIA is at address location \$5000 and PIA port B bits PB0 through PB7 are to be outputs. A possible software approach would be:

CLRA	Clear accumulator A insuring bit 2 contains a zero.
STAA \$5003	This stores a zero into bit 2 of the Control Register and selects the Data Direction Register.
LDAA #\$FF	Load accumulator A with all ones.
STAA \$5002	This makes PB0 through PB7 outputs.
LDAA #\$04	Puts a "1" into bit 2.
STAA \$5003	Stores a "1" into bit 2 of control register allowing data to be written to PB0 to PB7.
LDAA #\$0F	
STAA \$5002	This would put the actual bit pattern output "0001111" on the PB0 through PB7 lines.

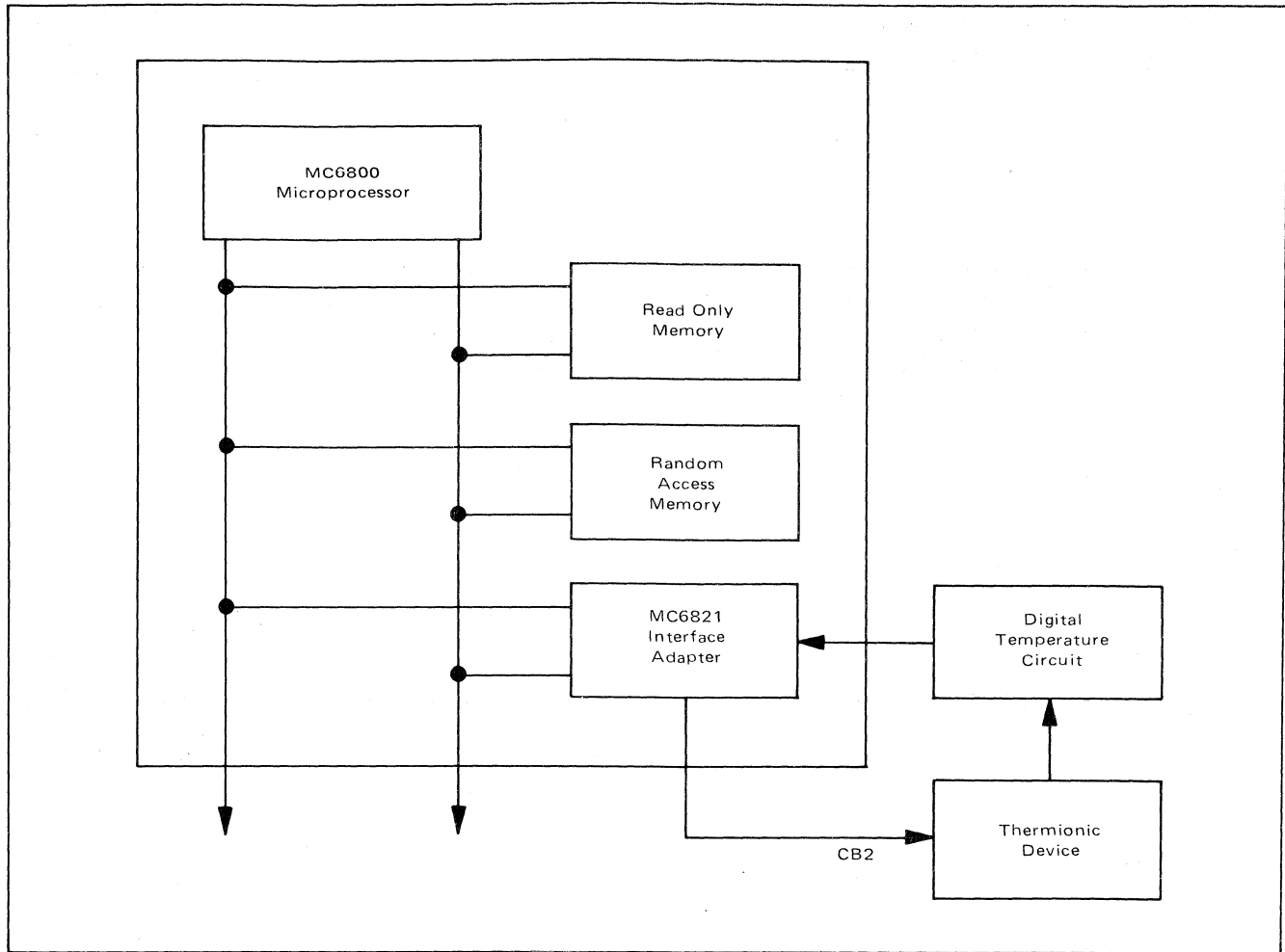


FIGURE 1. System Block Diagram

Circuit diagrams external to Motorola products are included as a means of illustrating typical semiconductor applications; consequently, complete information sufficient for construction purposes is not necessarily given. The information in this Application Note has been carefully checked and is believed to be entirely reliable. However, no responsibility is assumed for inaccuracies. Furthermore, such information does not convey to the purchaser of the semiconductor devices described any license under the patent rights of Motorola Inc. or others.

## Temperature Control System

The software which monitors the digital temperature data and decides if it is too high or too low is shown in Figure 2.

The upper and lower temperatures may be easily changed within the software for a variable temperature window. The Software Interrupt Command (SWI) causes a system interrupt if the temperature extends above or below the "window." In the software example, the temperatures were set for 27°C and 17°C. The software monitors the incoming temperature and goes to SWI if the temperature is equal to or greater than 27°C or equal to or less than 17°C. If a device such as an oven were to be turned on/off, the control bits CA2 (CB2) could be set accordingly to control the device. As shown in Figure 3, bits 3, 4 and 5 of the Control Register can be configured to Set/Reset CA2. For example, the instructions below would manipulate CA2.

To turn CA2 on:

```
LDAA #$3C      Load accumulator A with 0011 1100
                CA2 goes high
STAA $5001     Store accumulator A into the Control
                Register.
```

To turn CA2 off:

```
LDAA #$34      Load accumulator A with 0011 0100
                CA2 goes low
STAA $5001     Store accumulator A into the Control
                Register.
```

The data input to the MC6821 PIA is not necessarily restricted to digital temperature data. Any device which provides or accepts digital data can be interfaced to the MC6800 MPU through the MC6821 PIA. Manipulation of the software allows for a variety of applications.

NAM	TEMPS
CLRA STAA \$5003	Insures a '0' in bit 2. Stores a '0' in bit 2 of the Control Register which selects the Data Direction Register.
STAA \$5002	Stores all zeros into the Data Direction Register making PIA Port B. PBO to PB7 lines all inputs. This will input the digital temperature data from the temperature device.
LDAA #\$04 STAA \$5003	Puts a '1' in bit 2. Stores a '1' in bit 2 of the Control Register which selects the Output Register. The P0 to PB7 may be read at \$5002 for the data being applied to them.
LBL1/LDAA #\$27 LDAB \$5002	The upper temperatures limit. Load in the digital temperature data which is on the PBO to PB7 lines.
CBA BLE ALERT LDAA #\$17 LDAB \$5002	Compare the temperatures. If equal to or greater then 27 interrupt. Lower temperature limit. Input temperature from sense circuit.
CBA BGE ALERT	Compare the temperatures. If equal to or less than 17 go to location ALERT and SWI.
BRA LBL1 ALERT/SWI	Branch back if within temperature window. Software Interrupt

FIGURE 2. Temperature Control Software

**Determine Active CA1 (CB1) Transition for Setting Interrupt Flag IRQA(B)1 – (bit b7)**

- b1 = 0 : IRQA(B)1 set by high-to-low transition on CA1 (CB1).
- b1 = 1 : IRQA(B)1 set by low-to-high transition on CA1 (CB1).

**IRQA(B) 1 Interrupt Flag (bit b7)**

Goes high on active transition of CA1 (CB1); Automatically cleared by MPU Read of Output Register A(B). May also be cleared by hardware Reset.

**CA1 (CB1) Interrupt Request Enable/Disable**

- b0 = 0 : Disables IRQA(B) MPU Interrupt by CA1 (CB1) active transition.<sup>1</sup>
  - b0 = 1 : Enable IRQA(B) MPU Interrupt by CA1 (CB1) active transition.
1. IRQA(B) will occur on next (MPU generated) positive transition of b0 if CA1 (CB1) active transition occurred while interrupt was disabled.

b7	b6	b5	b4	b3	b2	b1	b0
IRQA(B)1 Flag	IRQA(B)2 Flag	CA2(CB2) Control		DDR Access	CA1 (CB1) Control		

**IRQA(B)2 Interrupt Flag (bit b6)**

CA2 (CB2) Established as Input (b5 = 0): Goes high on active transition of CA2 (CB2); Automatically cleared by MPU Read of Output Register A(B). May also be cleared by hardware Reset.

CA2 (CB2) Established as Output (b5 = 1): IRQA(B)2 = 0, not affected by CA2 (CB2) transitions.

**Determines Whether Data Direction Register Or Output Register is Addressed**

- b2 = 0 : Data Direction Register selected.
- b2 = 1 : Output Register selected.

**CA2 (CB2) Established as Output by b5 = 1**

(Note that operation of CA2 and CB2 output functions are not identical)

**CA2**

b5 = 1, b4 = 0, b3 = 0 : Read Strobe With CA1 Restore  
 CA2 goes low on first high-to-low E transition following an MPU Read of Output Register A; returned high by next active CA1 transition.

b5 = 1, b4 = 0, b3 = 1 : Read Strobe with E Restore  
 CA2 goes low on first high-to-low E transition following an MPU Read of Output Register A; returned high by next high-to-low E transition.

**CB2**

b5 = 1, b4 = 0, b3 = 0 : Write Strobe With CB1 Restore  
 CB2 goes low on first low-to-high E transition following an MPU Write into Output Register B; returned high by the next active CB1 transition.

b5 = 1, b4 = 0, b3 = 1 : Write Strobe With E Restore  
 CB2 goes low on first low-to-high E transition following an MPU Write into Output Register B; returned high by the next low-to-high E transition.

**Set/Reset CA2 (CB2)**

b5 = 1, b4 = 1, b3 = 0 : CA2 (CB2) goes low as MPU writes b3 = 0 into Control Register.

b5 = 1, b4 = 1, b3 = 1 : CA2 (CB2) goes high as MPU writes b3 = 1 into Control Register.

**CA2 (CB2) Established as Input by b5 = 0**

b5 = 0, b4 = 0, b3 = 0 : CA2 (CB2) Interrupt Request Enable/Disable

b3 = 0 : Disables IRQA(B) MPU Interrupt by CA2 (CB2) active transition.<sup>1</sup>

b3 = 1 : Enables IRQA(B) MPU Interrupt by CA2 (CB2) active transition.

1. IRQA(B) will occur on next (MPU generated) positive transition of b3 if CA2 (CB2) active transition occurred while interrupt was disabled.

b5 = 0, b4 = 0, b3 = 1 : Determines Active CA2 (CB2) Transition for Setting Interrupt Flag IRQA(B)2 – (bit b6)

b4 = 0 : IRQA(B)2 set by high-to-low transition on CA2 (CB2).

b4 = 1 : IRQA(B)2 set by low-to-high transition on CA2 (CB2).

FIGURE 3 – PIA Control Register Format

