BM

Brief Manual of HERA Application Board

With CORERIVER MiDAS1.0 Family

V1.1

October 2005

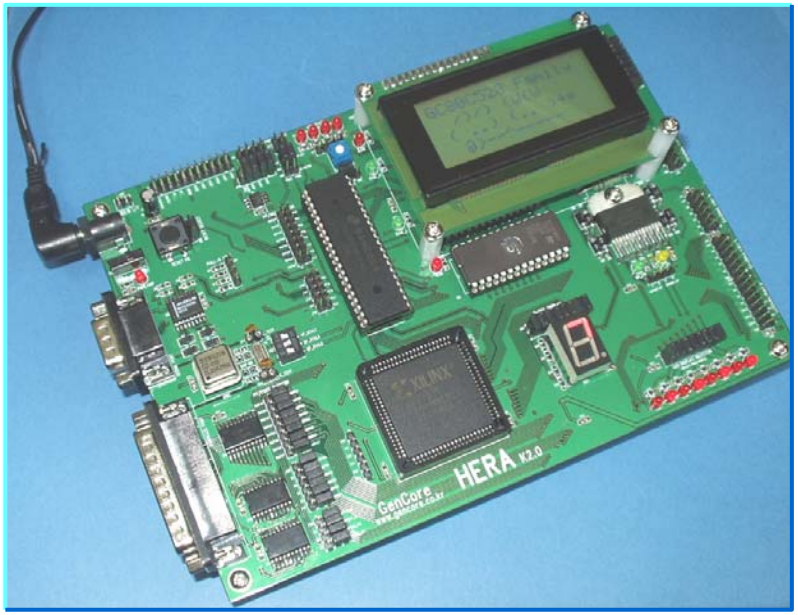
Contents

1. Introduction

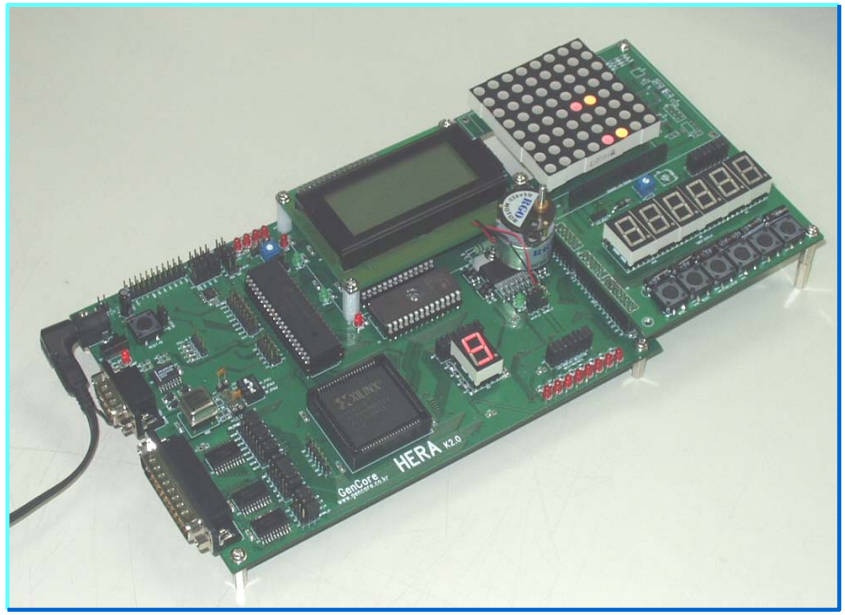
- What is the **HERA Application Board**?
  - ✓ HERA Application Board is designed as a low cost platform for learning the CORERIVER's Turbo MiDAS1.0 family and peripheral components.

- Configuration of **HERA Application Board**
  - ✓ Main Board
    - The MiDAS1.0 MCU: Its execution time is about 3 times faster than that of traditional 80C52.
    - External Data Memory : Max. 64KBytes SRAM (32kBytes x 2 ea)
    - External Program Memory : Max. 64KBytes EPROM
    - Serial Port and Parallel Port Communication
    - ADC Function Test : External 4-channel analog inputs, or internal temperature sensor (LM35DM)
    - External Ports and Display Applications

2. HERA Application Board

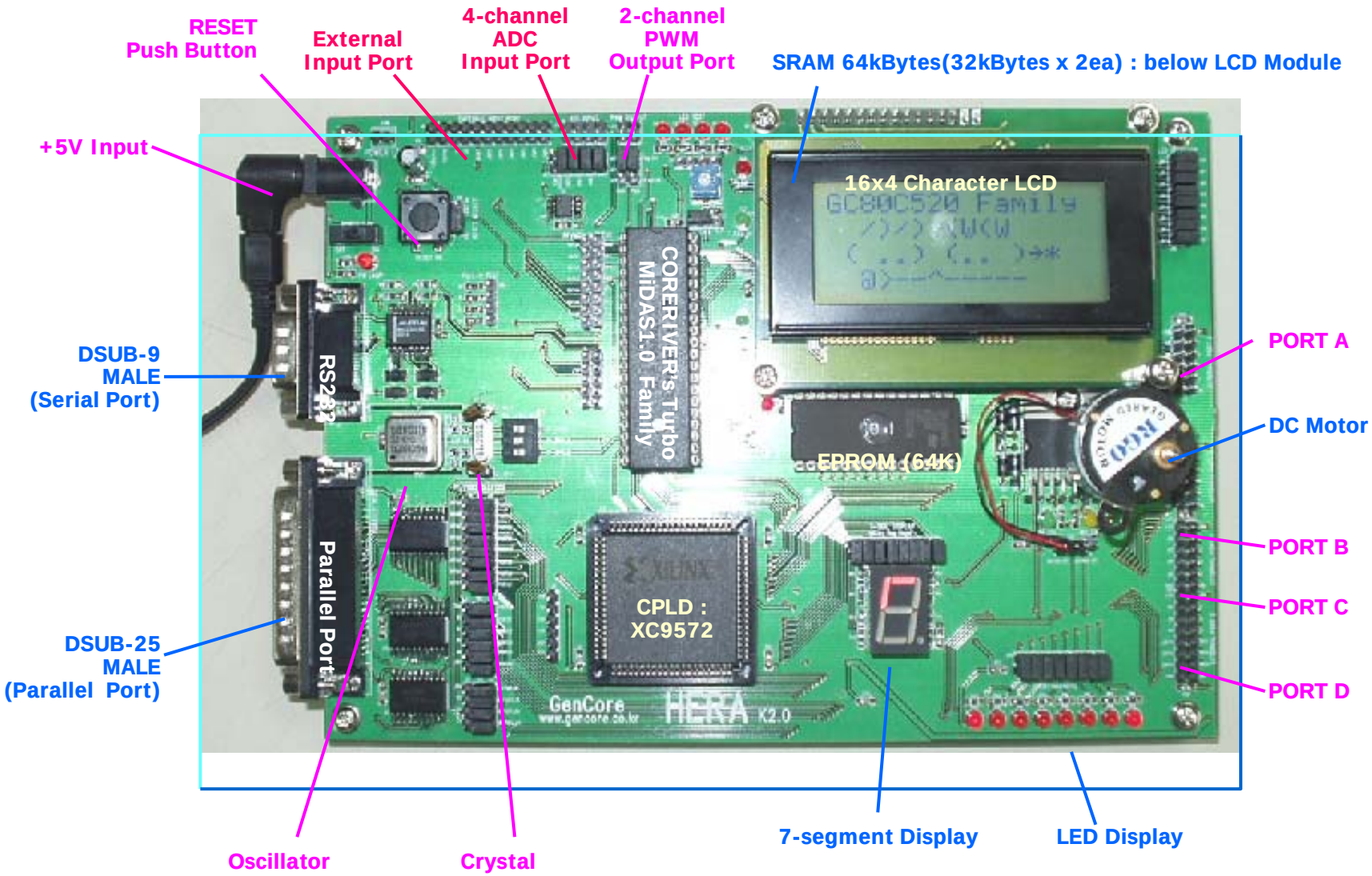


[Main Board]



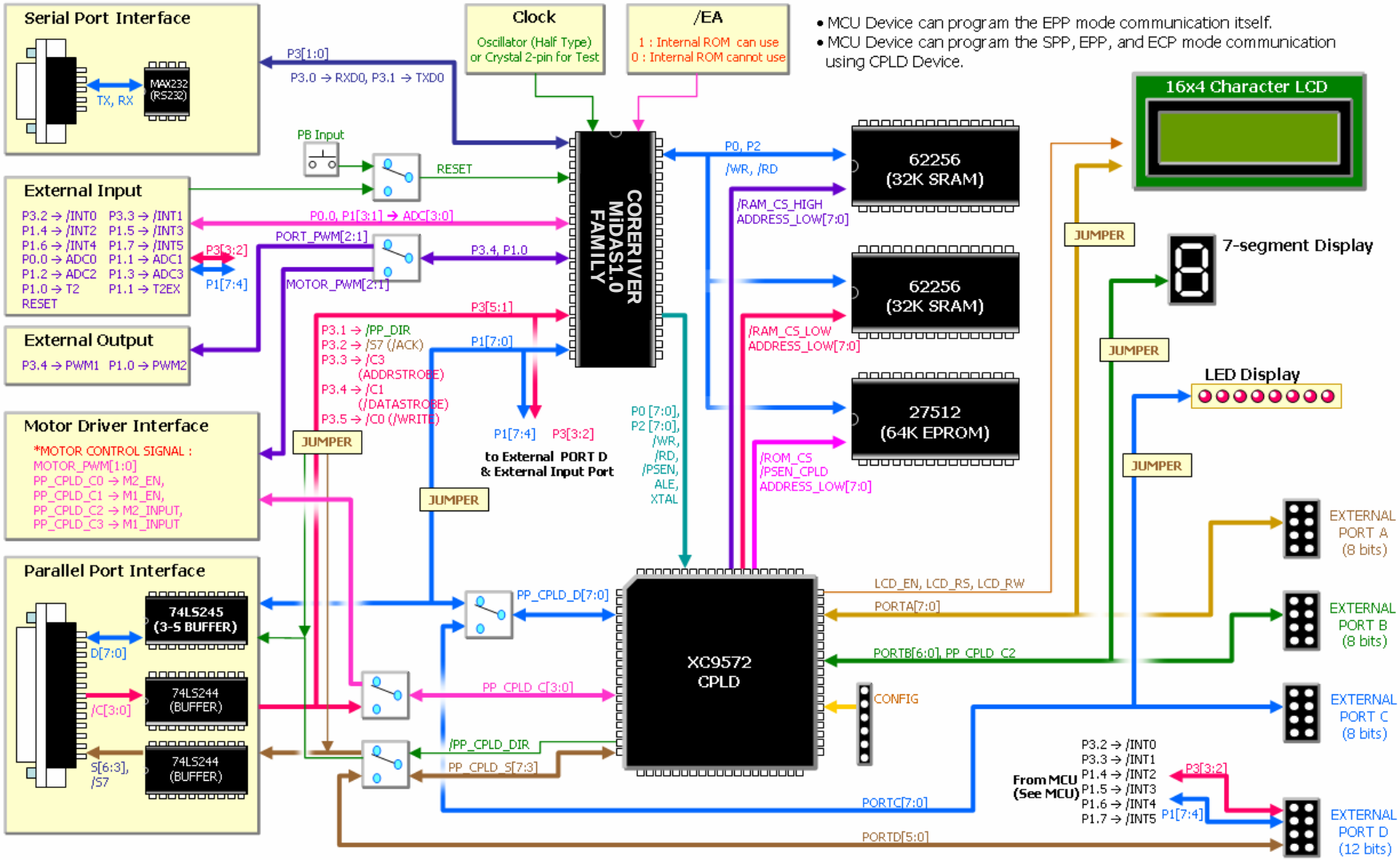
[Main Board with Sub Board]

3. Main Board

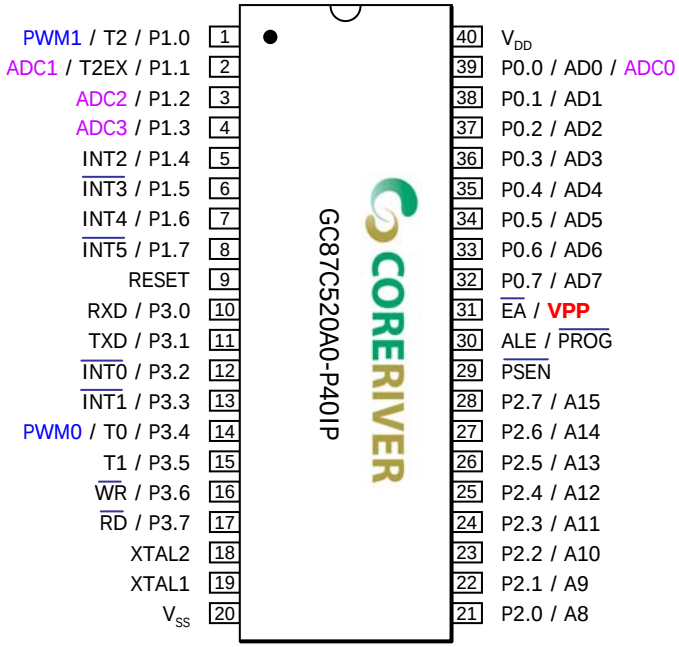


[Main Board]

4. Block Diagram of Main Board



Pin Configurations



[40-PDIP]

- **For more detail information, refer to the MCU Brief Manual "BM-MiDAS1.0-VXX.PPT".**

5. Address Map : Verilog Source

- Memory Map
  - ✓ User can reconfigure the address map in CPLD using Verilog HDL.

8'hFFXX	**: PORTA**
8'hFEXX	**: PORTB**
8'hFDXX	**: PORTC**
Data Memory (8'hFA00 ~ 8'hFCFF)	**: for Other Control BUS**
8'hF9XX	**: LCD_RW Setting**
8'hF8XX	**: LCD_RS Setting**
8'hF7XX	**: LCD_EN Setting**
Not-defined (8'hC0000 ~ 8'hF6FF)	
Data Memory (8'h0000 ~ 8'hBFFF)	

```
// File Name : HERA_K2.v
// Programmed by H. Jee
// Programmed Date            : Tue., May 13, 2003
// Lately Modified Date       : Thu., May 15, 2003
// Revision    : 0.1

module HERA_K2(
              P0,                              // input
              P2,
              WR_B,
              RD_B,
              PSEN_B,
              ALE,
              XTAL1,
              RAM_CS_HIGH_B,   // output
              RAM_CS_LOW_B,
              ADDRESS_LOW,
              ROM_CS_B,
              PSEN_CPLD_B,
              LCD_CON,
              PORTA,
              PORTB,
              PP_CPLD_D
);

// Declaration for Variables
// Input Variables
input [7:0] P0;
input [7:0] P2;
input WR_B;
input RD_B;
input PSEN_B;
input ALE;
input XTAL1;

// Output Variables
output RAM_CS_HIGH_B;
output RAM_CS_LOW_B;
output [7:0] ADDRESS_LOW;
output ROM_CS_B;
output PSEN_CPLD_B;

output [2:0] LCD_CON;
```

5. Address Map : Verilog Source (Cont'd)

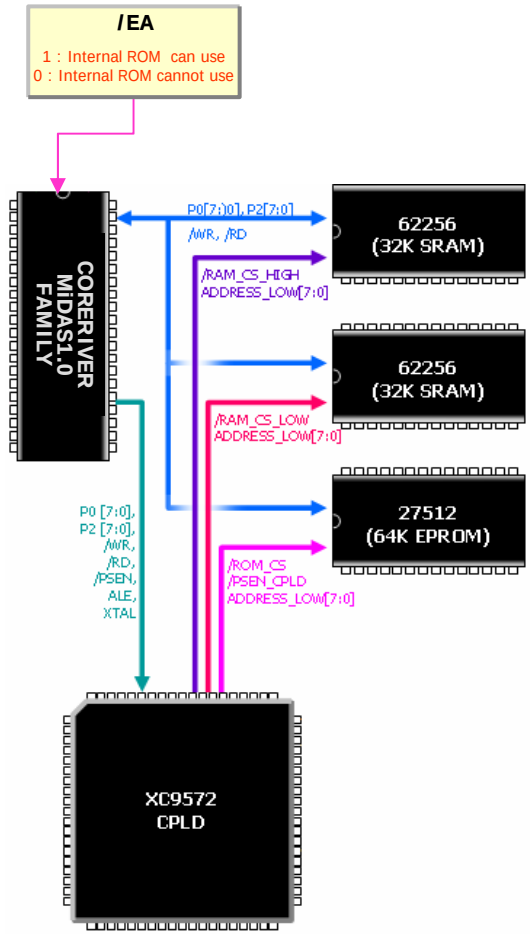
```
// External Port A & LCD 1-byte Data
output [7:0] PORTA;

// External Port B & 7-segment 1-byte Control
// & Parallel Port Data Bus 1 byte using Jumper
output [7:0] PORTB;

// External Port C, LED 1-byte Control
output [7:0] PP_CPLD_D;



// Declaration for Operations

// -ROM Read
assign ROM_CS_B = PSEN_B;
assign PSEN_CPLD_B = PSEN_B;

// -RAM Read & Write
//  -Address : 0000~7FFF (Low Byte)
assign RAM_CS_LOW_B = P2[7] | (WR_B & RD_B);
//  -Address : 8000~BFFF (High Byte)
assign RAM_CS_HIGH_B = ~P2[7] | P2[6] | (WR_B & RD_B);


// Low Address Latch using ALE & XTAL1 Signals
reg [7:0] ADDRESS_LOW;
reg [7:0] ADDRESS_HIGH;

reg [7:0] PORTA;
reg [7:0] PORTB;
reg [7:0] PP_CPLD_D;
reg [2:0] LCD_CON;
```

```
always @(posedge XTAL1)
              begin

              if (ALE)
              begin
                ADDRESS_LOW = P0;     // Low Address
                ADDRESS_HIGH = P2;    // for Address Mapping
              end

              if (~WR_B)
              begin

                if (ADDRESS_HIGH == 8'hF9)
                              //LCD_RW <= P0;
                              LCD_CON[0] <= P0[0];
                else if (ADDRESS_HIGH == 8'hF8)
                              //LCD_RS <= P0;
                              LCD_CON[1] <= P0[0];
                else if (ADDRESS_HIGH == 8'hF7)
                              //LCD_EN <= P0;
                              LCD_CON[2] <= P0[0];
                else if (ADDRESS_HIGH == 8'hFF)
                              begin
                              PORTA <= P0;
                              end
                else if (ADDRESS_HIGH == 8'hFE)
                              begin
                              PORTB <= P0;
                              end
                else if (ADDRESS_HIGH == 8'hFD)
                              begin
                              PP_CPLD_D <= P0;
                              end
                else
                              begin
                              end
                end
              end


// The End of Declaration for Operations

endmodule
```

6. Memory Interface

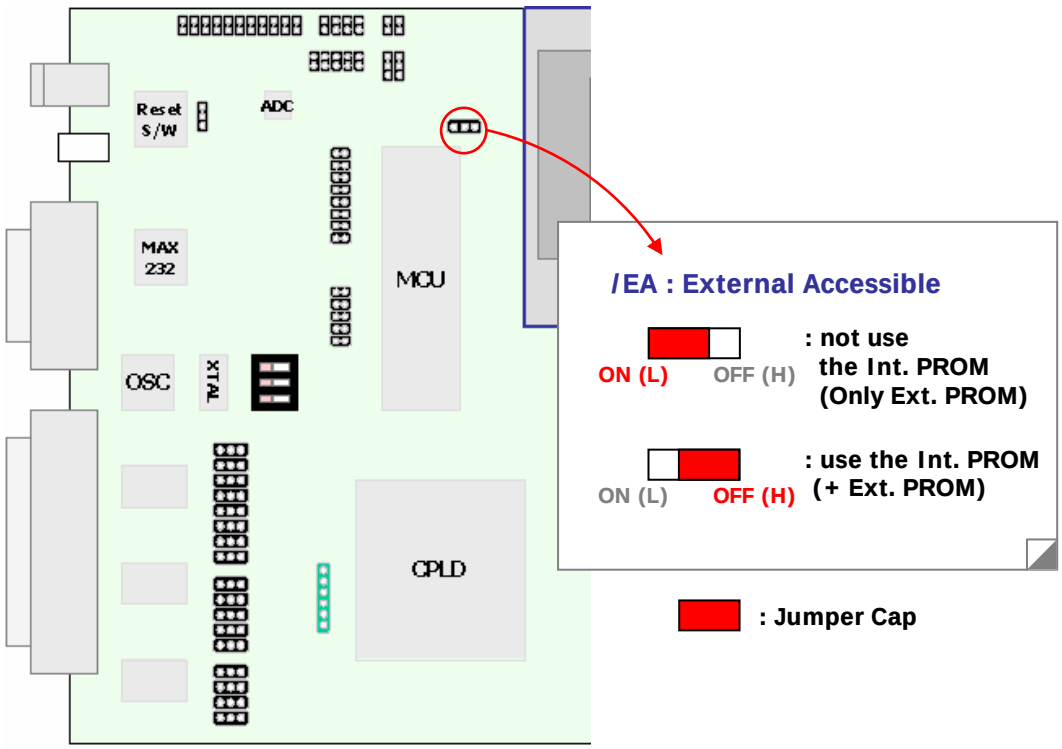
- Refer to Slide 6(Address Map : Verilog Source)
  - ✓ /EA of MCU → decide whether the internal program memory is used or not
    - /EA = 1 : internal program memory is used
    - /EA = 0 : internal program memory is not used

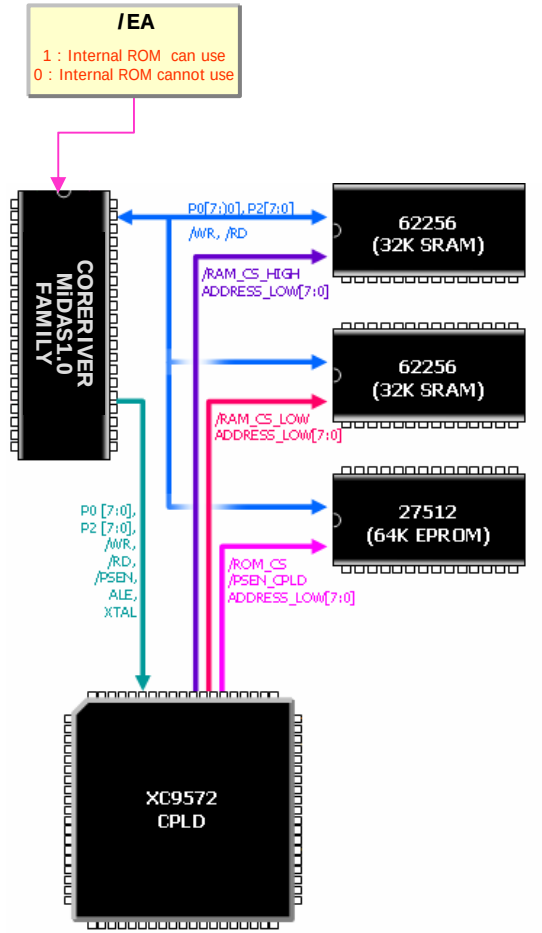
```
// -ROM Read
assign ROM_CS_B = PSEN_B;
assign PSEN_CPLD_B = PSEN_B;

// -RAM Read and Write
//  -Address : 0000~7FFF (Low Byte)
assign RAM_CS_LOW_B = P2[7] | (WR_B & RD_B);
//  -Address : 8000~BFFF (High Byte)
assign RAM_CS_HIGH_B = ~P2[7] | P2[6] | (WR_B & RD_B);


always @(posedge XTAL1)
             begin

             if (ALE)
             begin
               ADDRESS_LOW = P0;      // Low Address
               ADDRESS_HIGH = P2;     // for Address Mapping
             end
```



6. Memory Interface

- Board Configuration

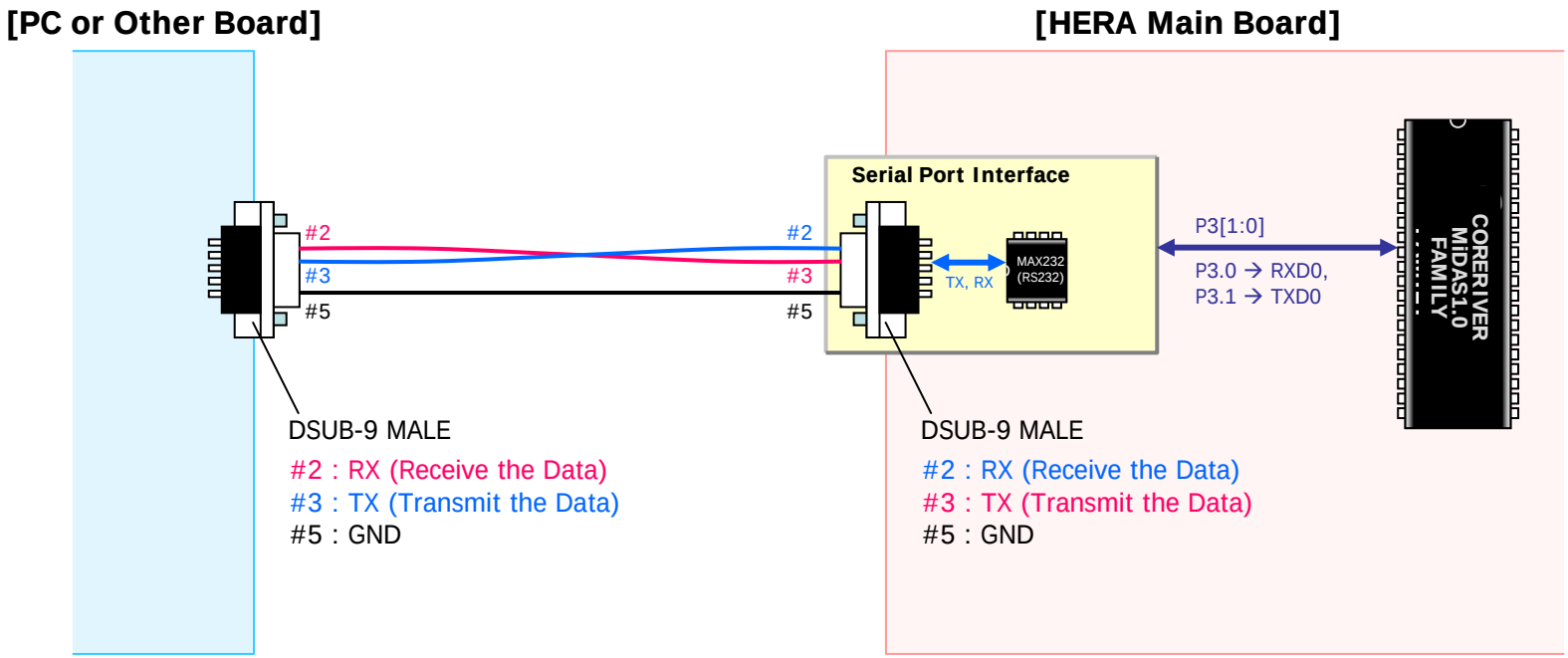




7. Serial Port Communication

- HERA uses the MAX232 device for serial communication.
- Serial port is controlled by three registers : SBUF, SCON, and PCON.

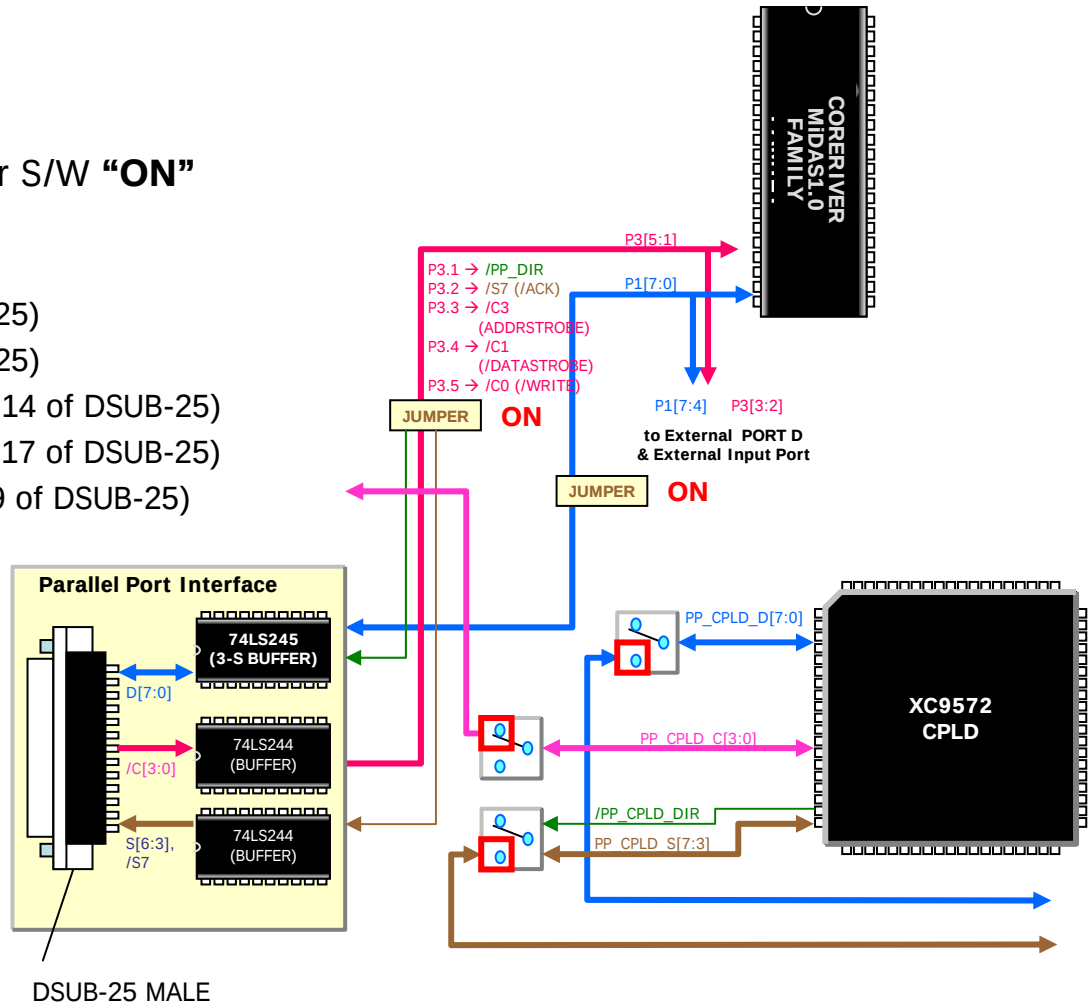
- For more detail information, refer to MiDAS1.0 family Reference Manual.



8. Parallel Port Communications : using MCU

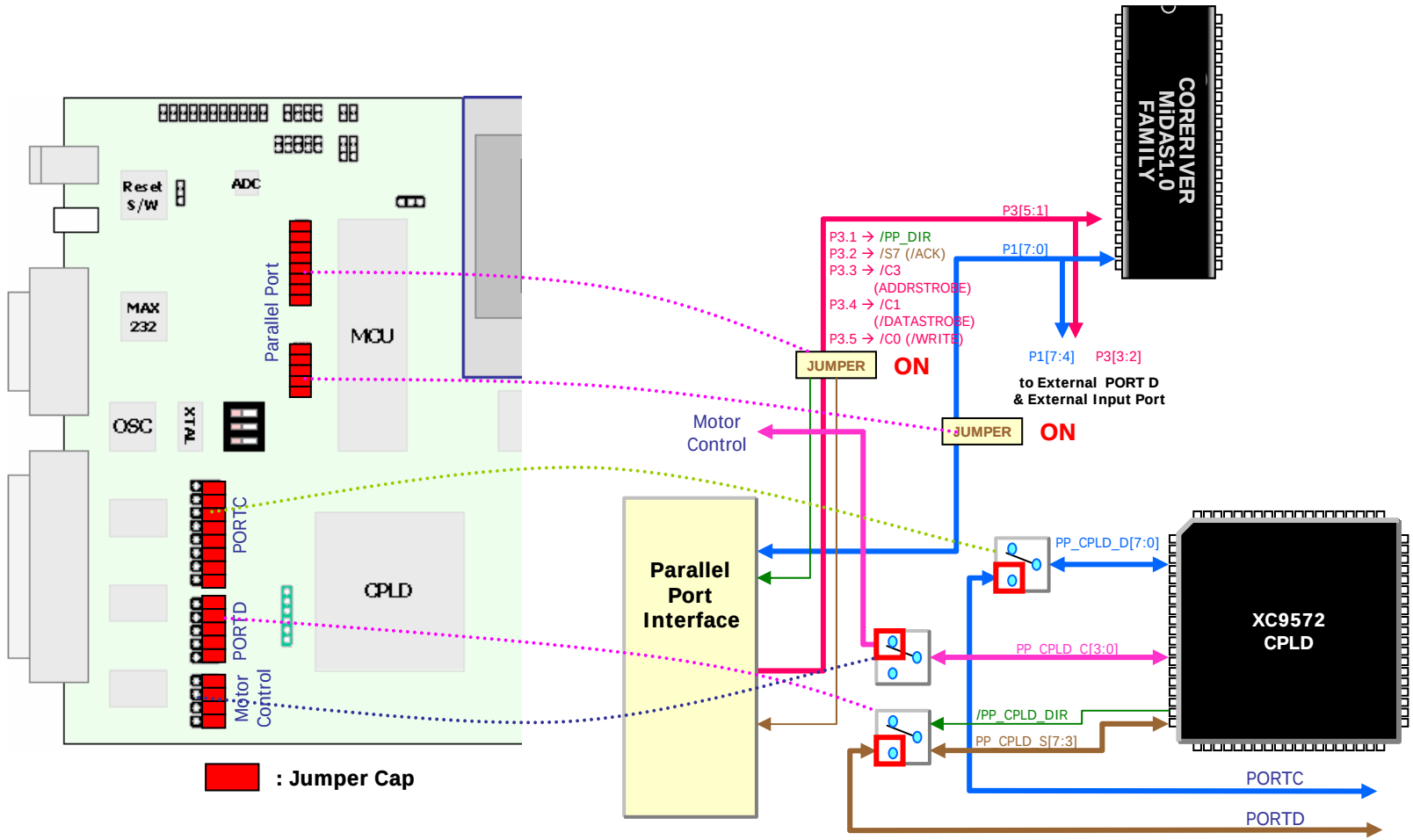
- Parallel Port
  - ✓ SPP (Standard Parallel Port)
  - ✓ EPP (Enhanced Parallel Port)
  - ✓ ECP (Extended Compatibility Port)

- Parallel Port – using MCU : Jumper S/W **"ON"**
  - ✓ EPP communications is possible.
  - ✓ Total 12 bits
    - /WRITE (#01 of DSUB-25)
    - /ACK (#11 of DSUB-25)
    - /DATASTROBE (#14 of DSUB-25)
    - /ADDRSTROBE (#17 of DSUB-25)
    - ADDR/DATA[7:0] (#02 ~ #09 of DSUB-25)



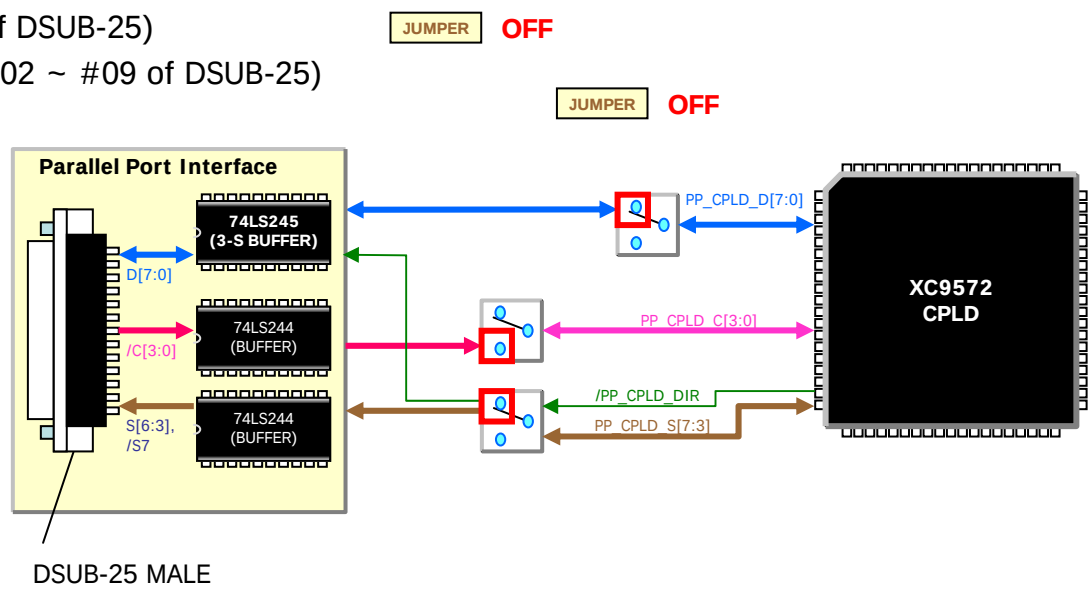
8. Parallel Port Communications : using MCU

- Board Configuration



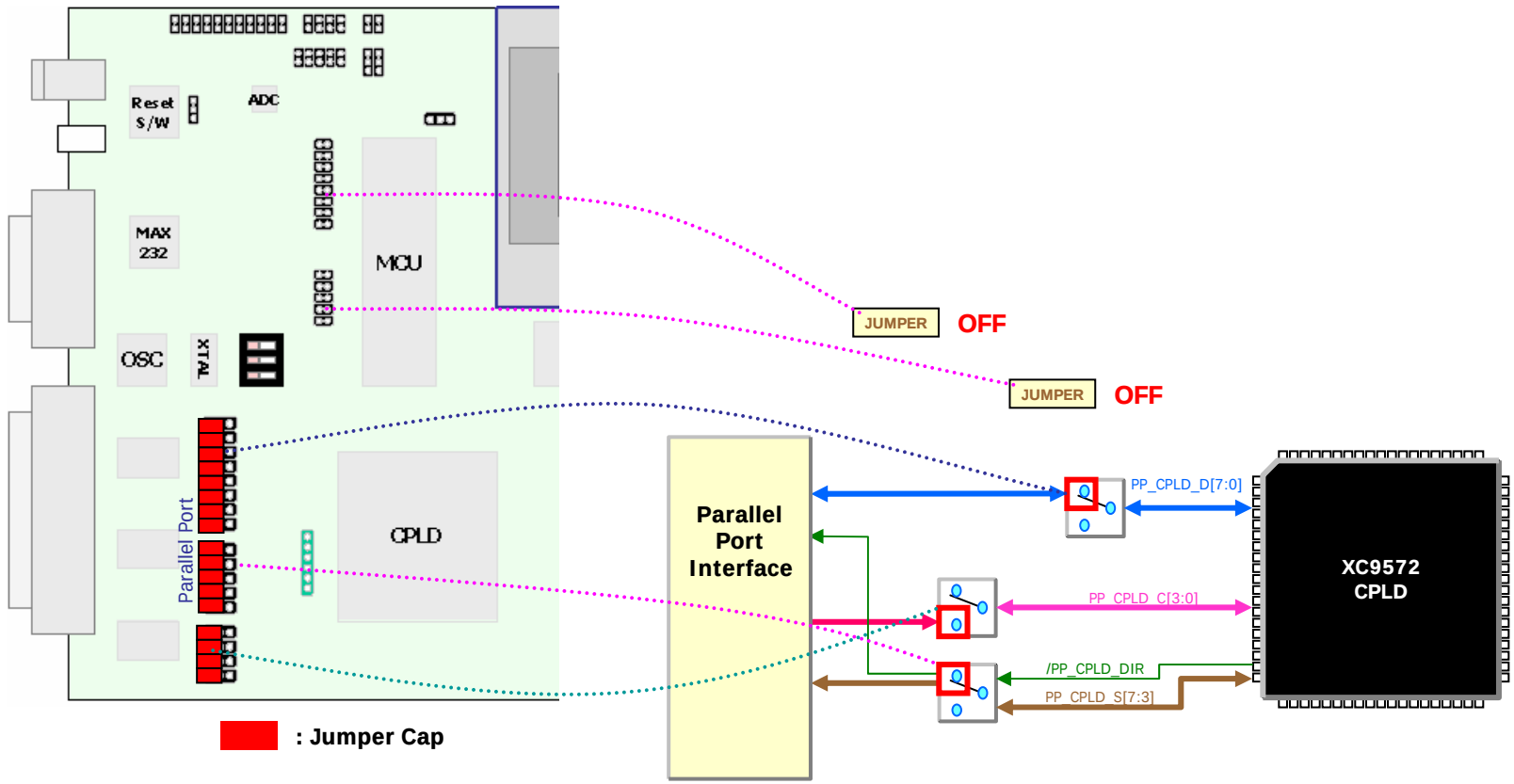
8. Parallel Port Communications : using CPLD

- Parallel Port – using CPLD : Jumper S/W “ON”
  - ✓ SPP, EPP and ECP communication is possible.
  - ✓ Total 17 bits
    - /STROBE (#01 of DSUB-25)
    - /ACK (#10 of DSUB-25)
    - BUSY (#11 of DSUB-25)
    - PERROR (#12 of DSUB-25)
    - SELECT (#13 of DSUB-25)
    - /AUTOFD (#14 of DSUB-25)
    - /FAULT (#15 of DSUB-25)
    - /INIT (#16 of DSUB-25)
    - /SELECTIN (#17 of DSUB-25)
    - ADDR/DATA[7:0] (#02 ~ #09 of DSUB-25)



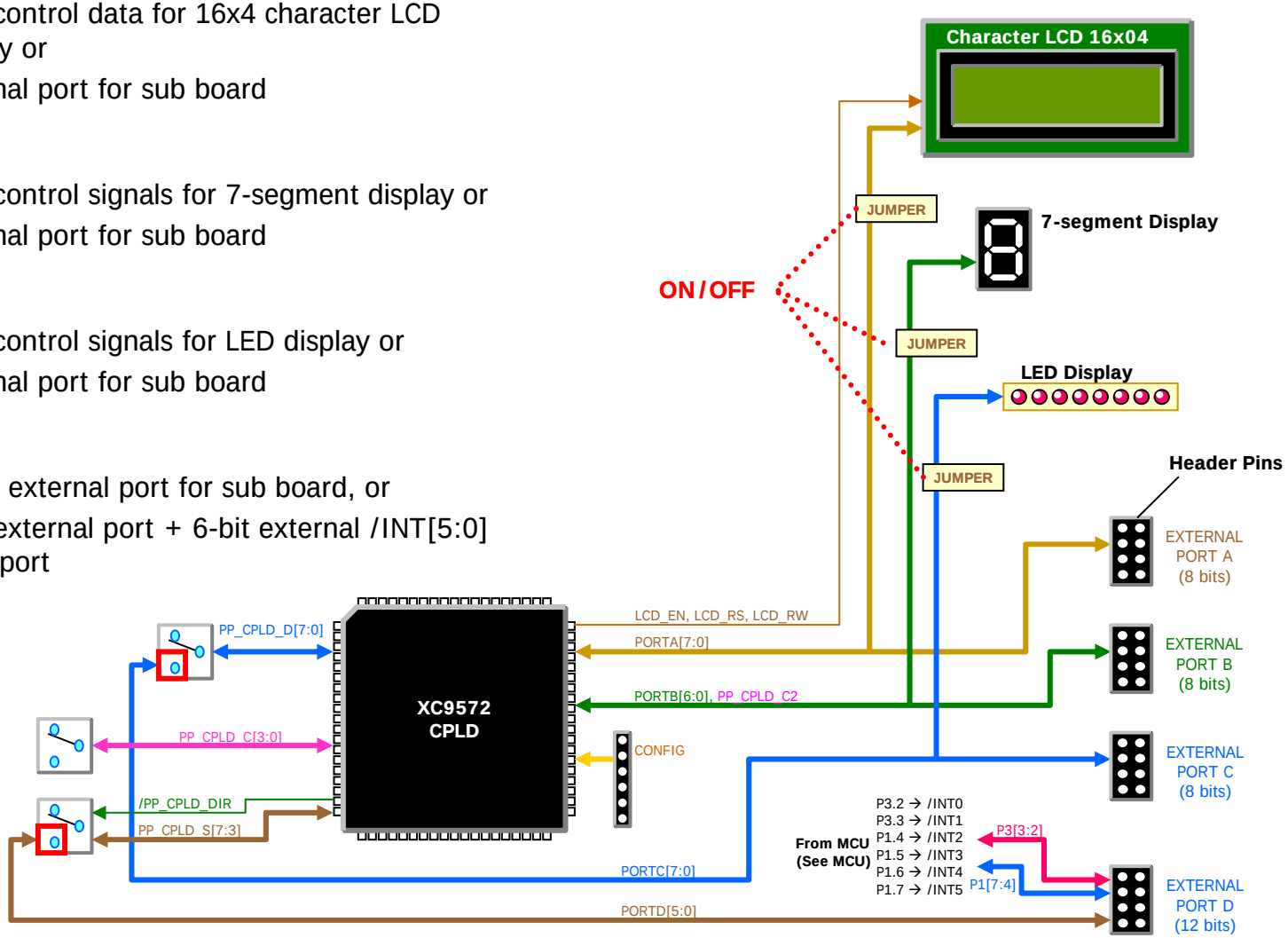
8. Parallel Port Communications : using CPLD

- Board Configuration



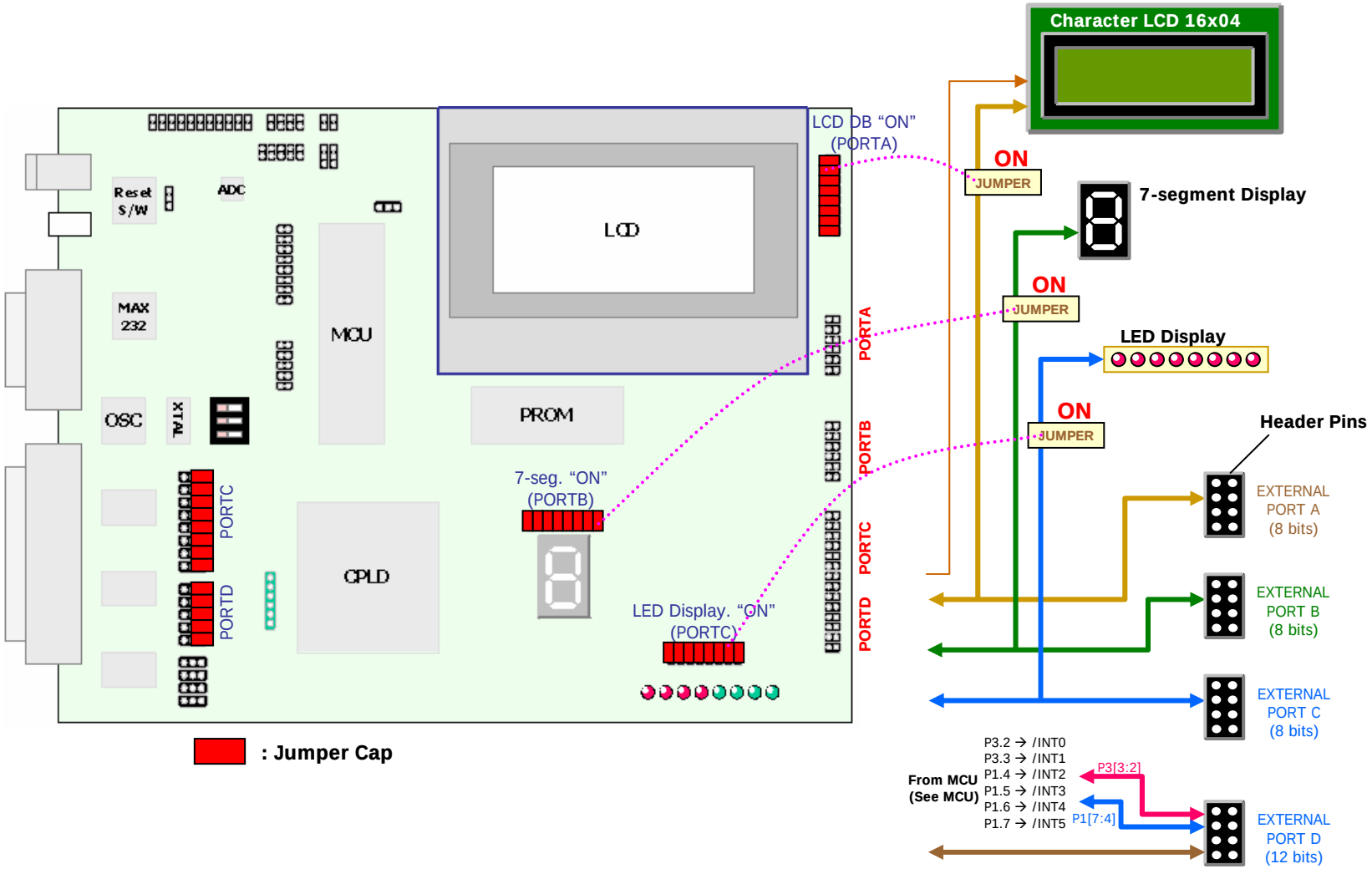
9. External Ports and Display Applications

- PORT A
  - ✓ 8-bit control data for 16x4 character LCD display or
  - ✓ External port for sub board
- PORT B
  - ✓ 8-bit control signals for 7-segment display or
  - ✓ External port for sub board
- PORT C
  - ✓ 8-bit control signals for LED display or
  - ✓ External port for sub board
- PORT D
  - ✓ 12-bit external port for sub board, or
  - ✓ 6-bit external port + 6-bit external /INT[5:0] input port



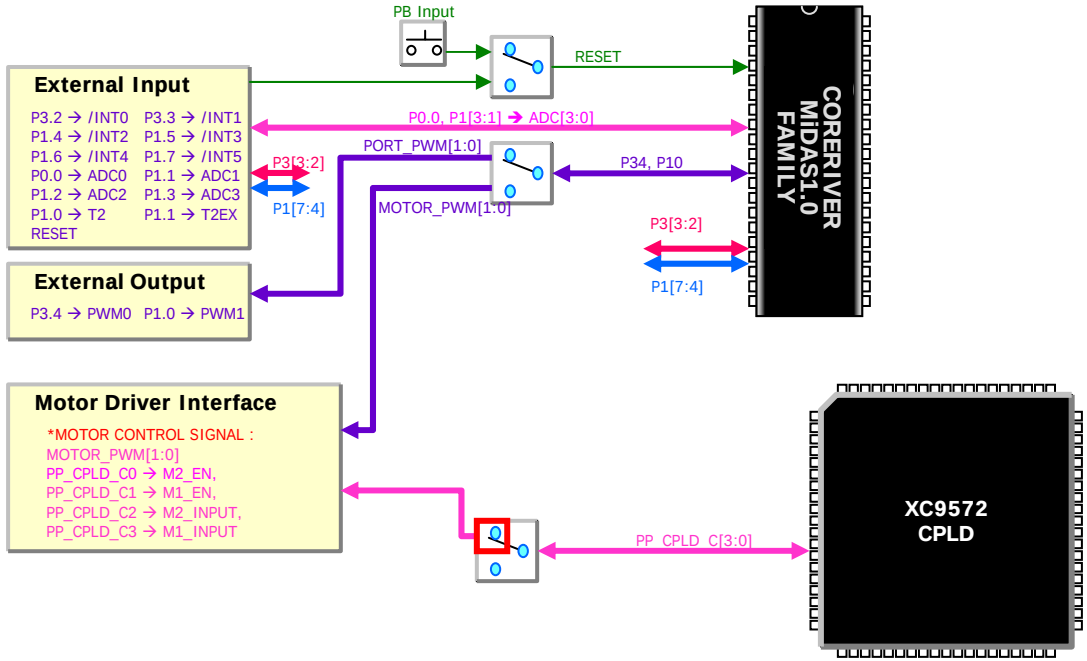
9. External Ports and Display Applications

- Board Configuration



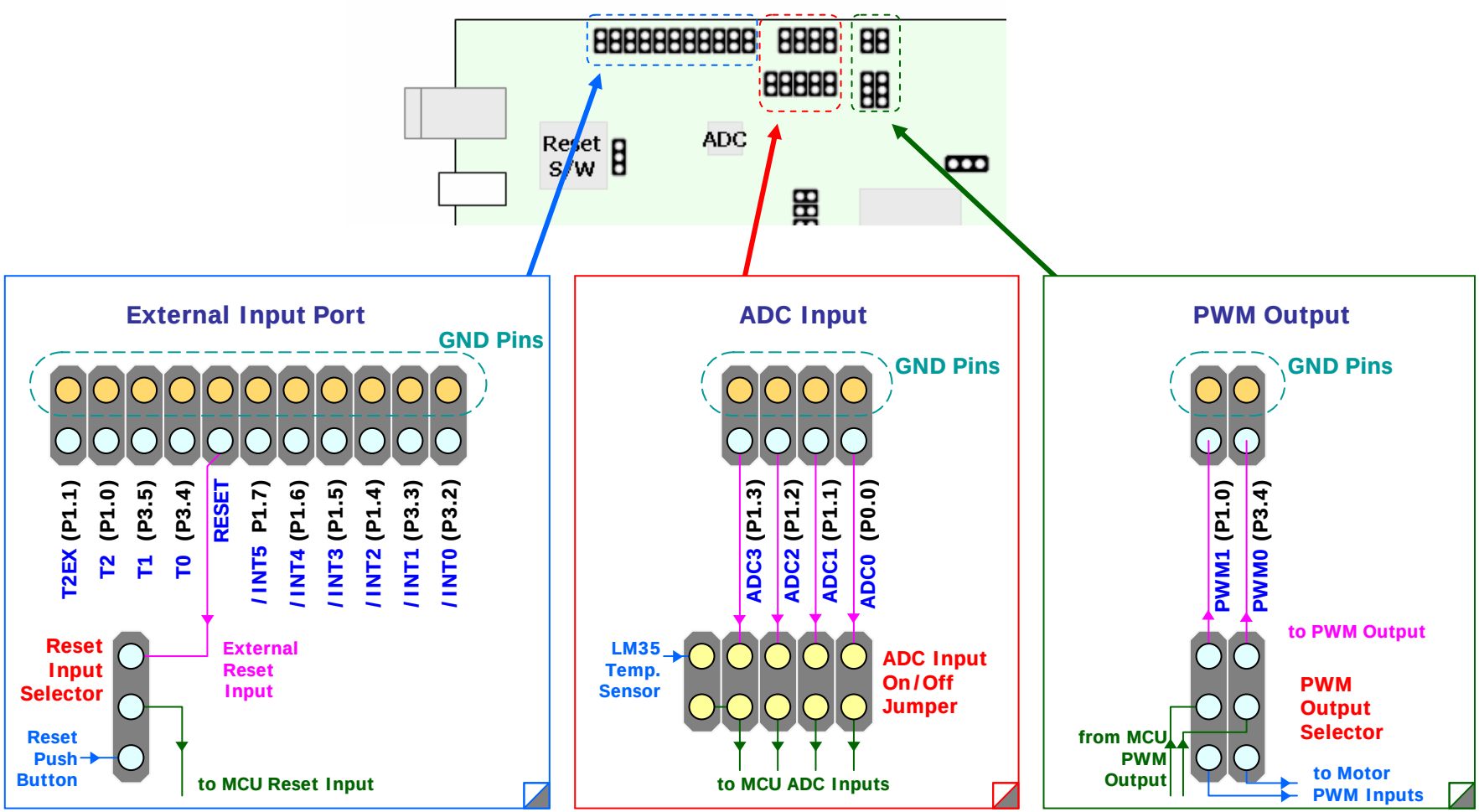
10. ADC Inputs, PWM Outputs, and Etc.

- MiDAS1.0 Family MCU supports
  - ✓ 4-channel ADC Inputs
  - ✓ 2-channel PWM Outputs
- ADC (Analog-to-Digital Converter)
  - ✓ External Input Ports for 4-channel ADC (ADC[3:0])
  - ✓ Temperature sensor (LM35) supports the analog output for ADC input test (ADC3).
- PWM (Pulse Width Modulation)
  - ✓ External Output Ports for 2-channel PWM (PWM[1:0])
  - ✓ It can use the PWM control signals to control DC motors.
  - ✓ PWM Control Signals control the velocity of DC motors.



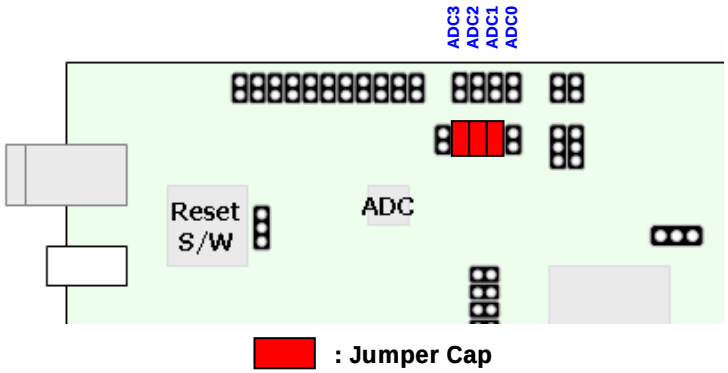
10. ADC Inputs, PWM Outputs, and Etc.

- External Input and Output Header Pins



10. ADC Inputs, PWM Outputs, and Etc.

Board Configuration for ADC Inputs

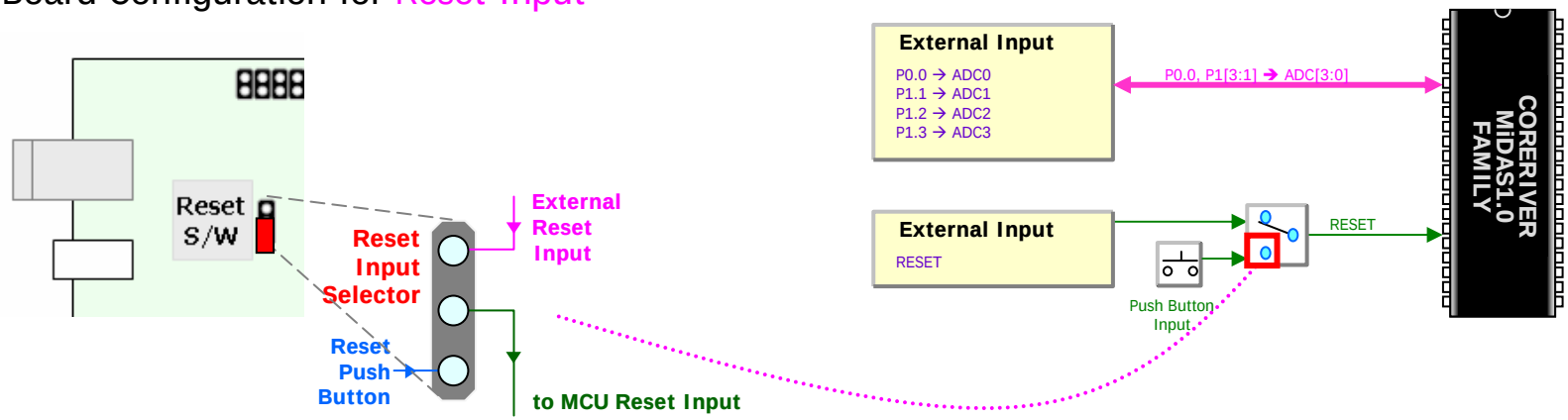


• Note

1. Please, turn off Pull-up Resistor to use ADC channels.
   (ex: MOV P0SEL, 0xFF)

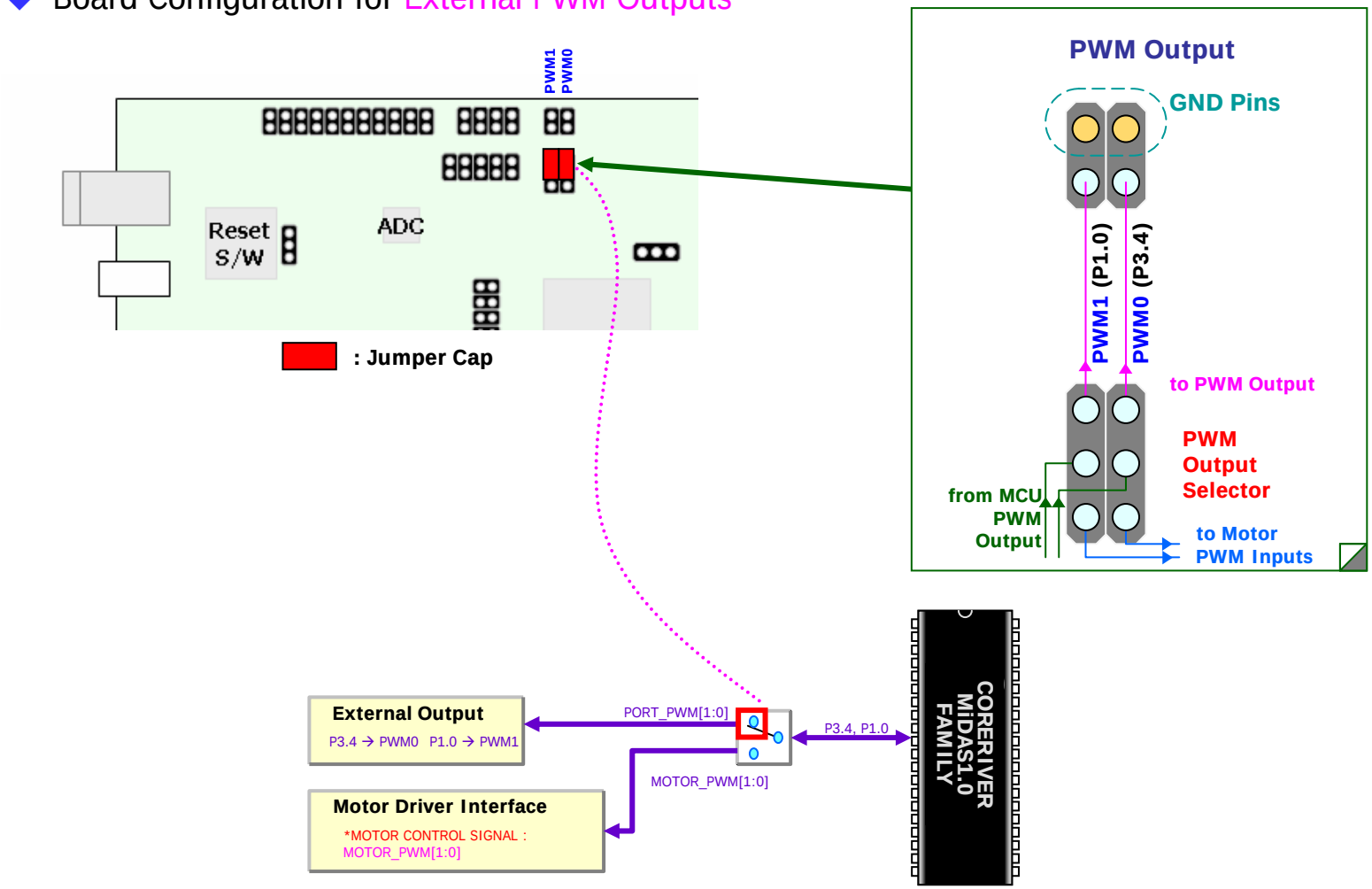
2. When you use an external (Data or Program) memory, ADC0 input channel cannot be used.

Board Configuration for Reset Input



10. ADC Inputs, PWM Outputs, and Etc.

- Board Configuration for External PWM Outputs



10. ADC Inputs, PWM Outputs, and Etc.

- Board Configuration for Motor Control