Embedded Development Platform
EDP Baseboard EDP-BB-4A User Manual
## Contents

1. **The EDP System**  
   1.1 Introduction ................................................................. 3  
   1.1.1 EDP Baseboard .......................................................... 3  
   1.1.2 Reusable Components .................................................. 3  
   1.1.3 Bread-Boarding Platform ............................................. 3  
   1.2 EDP Modules Available Now ........................................... 4  
   1.3 Basic EDP Concepts ....................................................... 4  
   1.3.1 Standardised Signal Set for Embedded Microcontrollers ....... 5  
   1.3.2 Grouping of Signals on the EDP Connectors ..................... 6  
   1.3.3 EDP Signal Names ........................................................ 8  
   1.4 The EDP Virtual CPU Concept ......................................... 9  
   1.4.1 Example of Real CPU to EDPCON Mapping ......................... 11  
   1.5 Inter-Module Communication ........................................... 14  
   1.6 Inter-EDP System Communications .................................... 14

2. **Using the EDP Baseboard**  
   2.1 EDP Connectors ............................................................. 15  
   2.2 EDP Baseboard User Options Placement ............................. 16  
   2.3 EDP Baseboard Component Placement ................................ 17  
   2.4 EDP I/O Pin Headers ....................................................... 18  
   2.5 Grounding Arrangements ................................................ 19  
   2.6 Positive Supplies .......................................................... 19  
   2.6.1 Logic Supplies ........................................................... 19  
   2.6.2 Analogue Supply ......................................................... 20  
   2.7 Limits and Restrictions ................................................... 20  
   2.8 EDP Control Busses ....................................................... 21  
   2.8.1 I²C Busses ................................................................. 21  
   2.9 CAN Network ............................................................... 23
1. The EDP System

1.1 Introduction

1.1.1 EDP Baseboard

The EDP Baseboard (or “motherboard”) consists of 4 ‘stations’ with the minimum configuration of the motherboard with a single plug-in processor module. All 4 stations are identical, and there are many permutations of CPU modules and Application modules possible. Even with just the minimum configuration of Motherboard and CPU module for example, you can easily run a web-server through the standard onboard Ethernet connection. There are various application modules; we have introduced an initial starter range consisting of basic digital and analogue I/O, a motor control module and a communications module. The more advanced user will discover that it is possible to run more than one processor module on the motherboard in a Master and Slave configuration.

The motherboard is an Extended Euro card size (220 x 100 mm) fitted with rubber feet to lay flat on the bench, but able to be used in a standard rack system. Add a 64-way DIN (RS 381-8696) connector and you can plug the EDP into a backplane. Connectors for four module stations are supplied, arranged to ensure correct module fitting. There are also fitted +3.3V and +5V voltage regulators, a back-up battery, an RJ45 Ethernet connector, a mini-USB connector, +12 volt power-supply jack, I/O breakout header and eight DIP switches ported onto the system I²C bus.

The DIP switches allow the user software running on a processor module to read a configuration setting, enabling I/O ports to be set up correctly, for example, or for CAN or TCP/IP addresses to be set. Depending on the capability of the particular processor module in use, up to three I²C buses and two CAN networks are available. Many of the application modules use an I²C bus for primary communication with the processor providing maximum flexibility. Some processor chips will require +5 volts, others +3.3 volts. A factory link on the module selects the correct supply from the connector. This supply is linked to a further connector pin on all the other module stations providing a correct voltage reference or bus pull-up for the application modules.

1.1.2 Reusable Components

The EDP baseboard is designed to be used and reused with new CPU and application modules being introduced on a regular basis. Its robust design has been rigorously tested, and every effort has been made at the design stage to protect the EDP from the most common human errors: the motherboard will have a significantly longer life than the average development board and is suitable for use in specialist one-off and low-volume products. Typical applications might be industrial controllers, scientific instrument controllers, data-logging and remote monitoring. For these reasons the EDP will prove attractive to all design engineers looking for a cost effective solution which allows them to significantly improve their development process and thus deliver products in reduced time. Design engineers, consultants, educators and trainers will quickly realise the benefits and recognise the potential of the development platform modules system as an effective solution.

1.1.3 Bread-Boarding Platform

With the difficulty in applying traditional “bread-boarding” techniques to today's tiny SMT components, evaluating new active devices has become major problem. There is usually no alternative to creating a special “try-out” PCB using rapid PCB production houses just to get a new device up and running.
The EDP has been designed to host such experimental and trial designs, providing “clean” +5 and +3V3 supplies and instant access to a range of standard microcontrollers and I/O blocks and devices. The design information necessary to allow you to create your own module for experimenting with new devices is available free of charge but in many cases, RS will already have such a module available to save you the effort.

The EDP represents the start of a continuous launch process which will see the introduction of new processor and application modules on a monthly basis.

### 1.2 EDP Modules Available Now

- **Processor Module:** ST Microelectronics STR912
- **Processor Module:** Infineon XC167
- **Application Module:** Analogue Input
- **Application Module:** Digital Input/Output
- **Application Module:** Brushed DC Motor Control
- **Application Module:** Basic Communications

### 1.3 Basic EDP Concepts

The EDP allows microcontrollers and I/O devices to communicate through a standardised interface. To some extent this interface is analogous to PC104 or STE busses where a connector pin-out is defined that allows the interconnection of address and data-bus connected devices. Such busses tend to include only power line, data and address busses plus control signals such as chip selects and interrupt request lines.

```
<table>
<thead>
<tr>
<th>Pin</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>I/OCK</td>
<td>GND</td>
</tr>
<tr>
<td>2</td>
<td>D7</td>
<td>+5V</td>
</tr>
<tr>
<td>3</td>
<td>D6</td>
<td>+5V</td>
</tr>
<tr>
<td>4</td>
<td>D5</td>
<td>IRQ9</td>
</tr>
<tr>
<td>5</td>
<td>D4</td>
<td>-5V</td>
</tr>
<tr>
<td>6</td>
<td>D3</td>
<td>DRQ2</td>
</tr>
<tr>
<td>7</td>
<td>D2</td>
<td>-12V</td>
</tr>
</tbody>
</table>
| 8   | D1   | ENDXFR#
| 9   | D0   | -12V|

Pins D and C are defined in the table below.

<table>
<thead>
<tr>
<th>Pin</th>
<th>A</th>
<th>B</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>GND</td>
<td>GND</td>
</tr>
<tr>
<td>1</td>
<td>MEMS16#</td>
<td>SBBIE#</td>
</tr>
<tr>
<td>2</td>
<td>I/OCS16#</td>
<td>LA23</td>
</tr>
<tr>
<td>3</td>
<td>IRQ10</td>
<td>LA22</td>
</tr>
<tr>
<td>4</td>
<td>IRQ11</td>
<td>LA21</td>
</tr>
<tr>
<td>5</td>
<td>IRQ12</td>
<td>LA20</td>
</tr>
<tr>
<td>6</td>
<td>IRQ15</td>
<td>LA19</td>
</tr>
<tr>
<td>7</td>
<td>IRQ14</td>
<td>LA18</td>
</tr>
<tr>
<td>8</td>
<td>DACKr1#</td>
<td>LA17</td>
</tr>
<tr>
<td>9</td>
<td>DRQ0</td>
<td>MEMR0</td>
</tr>
<tr>
<td>10</td>
<td>DACKr0#</td>
<td>MEMR0</td>
</tr>
<tr>
<td>11</td>
<td>DRQ5</td>
<td>SD8</td>
</tr>
<tr>
<td>12</td>
<td>DACKr#</td>
<td>SD9</td>
</tr>
<tr>
<td>13</td>
<td>DRQ6</td>
<td>SD10</td>
</tr>
<tr>
<td>14</td>
<td>DACKr#</td>
<td>SD11</td>
</tr>
<tr>
<td>15</td>
<td>DRQ7</td>
<td>SD12</td>
</tr>
<tr>
<td>16</td>
<td>+5V</td>
<td>SD13</td>
</tr>
<tr>
<td>17</td>
<td>MASTER#</td>
<td>SD14</td>
</tr>
<tr>
<td>18</td>
<td>GND</td>
<td>SD15</td>
</tr>
<tr>
<td>19</td>
<td>GND</td>
<td>GND/KEY</td>
</tr>
</tbody>
</table>

For microcontroller systems, such a collection of signals is of very limited use, especially for single-chip CPUs that use no external bus. It also takes no account of the specialist pin functions available on microcontrollers such as CAN, I2C, SPI, signal measurement and signal generation peripherals.
1.3.1 Standardised Signal Set for Embedded Microcontrollers

The EDPCON1 and 2 connectors thus defines a set of signals on a standardised format that are relevant to typical 8, 16 and 32-bit microcontrollers. In addition to address bus, data bus and chip select signals, they include three \textit{I}^2\textit{C} channels, two \textit{CAN} channels, groups of pins able to create interrupts in response to external events, groups of pins able to create pulse-trains, others dedicated to motor control, \textit{I}^2\textit{S}, memory cards and many other common microcontroller \textit{I/O} types.

All of these signals are contained within two 0.8mm dual-row connectors of 140 and 100 pins each.
1.3.2 Grouping of Signals on the EDP Connectors

The EDPCON1 and 2 connector specification divide the total available 240 pins into groups or regions of similar characteristics, as shown below:

1.3.2.1 EDPCON1 Connector I/O Regions

EDPCON1 carries both analogue and digital signals. The analogue signals are grouped together in a “quiet zone”.

---

**Analog Reference**, **Analog** channels region, **Analog Ground**, **General I/O pins that have no interrupt capability**, **General I/O pins that have an interrupt capability**, **General I/O pins that have no interrupt capability or which form an external multiplexed address and data bus**, **General I/O pins that have an interrupt capability and are able to generate pulse trainss or timed edge transitions** or **capture/compare capture pins**, **IO pins that are used for UART RX and TX**, **IO pins that are used for Ethernet PHYs**, **IO pins with I2C functionality**, **IO pins with CAN functionality**

---

© Electrocomponents plc
Vsn 1.1
Page 6
1.3.2.2 EDPCON2 Connector Regions

EDPCON2 carries mainly bus signals such as \( \text{I}^2\text{C}, \) SPI, CAN and the multiplexed 16-bit external bus from the CPU module.
1.3.3  EDP Signal Names

The generic signals present on the connectors have names which indicate their primary and secondary functions.

1.3.3.1  EDPCON1 Signal Description

ANx: Analogue signals
VAGND: Analogue ground, referenced to CPU and Analogue application analogue signal grounds
GPIOx: Pins that can only be set to 1 or 0 by a CPU instruction. It has no special or alternate function.
GPIOx_MClxxx: Pins that have basic I/O function like “GPIOx” but which also form an SM/MMC card interface
GPIOx_I2S_XXX: Pins that have basic I/O function like “GPIOx” but which also form an I2S interface.
IRQx_GPIOx X I2C_INT: Pins that are used by the three I\(^2\)C busses to request a CPU interrupt. **Note:** IRQ_GPIO16_CNTRL_I2C_INT should always be reserved for use by the I\(^2\)C CNTRL I2C bus.
CPU_DACx_GPIOx: Pins where CPUs with true digital to analogue converter outputs are always connected. Alternatively, PWM will be available if there is no DAC.
EVMx_GPIOx: Pins which have basic I/O function but which also can measure timed events, pulse times and durations e.g. CAPCOM input.
GPIOx_ADx: Pins with basic I/O function but which also can form a multiplexed address and data bus.
EVMx_GPIOx: Pins which have basic I/O function but which also can generate events like timed pulses and transitions e.g. CAPCOM output.
EVM2_GPIO41_CAPADC: Pins which have basic I/O function but which also can measure pulse times and durations e.g. CAPCOM input. If the CPU supports the triggering of ADC readings on an edge, the function will be on this pin.
ASC0_RX_TTL: Logic level connection to CPU module’s serial port 0 receive pin.
ASC0_TX_TTL: Logic level connection to CPU module’s serial port 0 transmit pin.
ASC1_RX_TTL: Logic level connection to CPU module’s serial port 1 receive pin.
ASC1_TX_TTL: Logic level connection to CPU module’s serial port 1 transmit pin.
ASC1_TX_TTL_ASC0_DTR: If CPU supports DTR function on ASC0, the function is available here.
ASC1_RX_TTL_ASC0_DSR: If CPU supports DSR function on ASC0, the function is available here.
EVM_GPIOx_ASC0_xTS: Event measurement, general I/O and ASC0 RTS and CTS functions, where available.
SPI_XXXX: Pins associated with SPI function, where supported by CPU module.
ETH_xxx: Pins connected to an Ethernet PHY on CPU module, where available.
I2C_GEN1_SDA/SCL: Pins connected to CPU’s I\(^2\)C channel 1
MOTOR_XXXX: Pins required for driving three-phase AC and DC brushless motors, including inputs for Hall sensors and tachometers or other speed-related signals.
EMRG_TRP: Emergency stop/trip function for motor control.
CAN1_RX/TX: Logic level connection to CPU module’s second CAN module (where fitted).
VCC_CM: Peripheral operating voltage of CPU module currently fitted.
+3V3: +3V3 supply from baseboard voltage regulator
+5V: +5V supply from baseboard voltage regulator
+12V: Raw +12V from power input to baseboard
12VGND: Ground connection to power supply.
SGND: Digital logic ground (connects to 12VGND at star point in baseboard
3V3 Vbatt: Permanent +3V3 supply from Lithium cell on baseboard (where fitted)

1.3.3.2 EDPCON2 Signal Description

#RESIN: Reset input to CPU module
#RESEOUT: Reset out signal from CPU module (where available)
l2C_GEN0_SDA/SCL: Secondary i2C bus data and clock (where available)
SGND: Digital logic ground (connects to 12VGND at star point in baseboard)
Axx_ADxx: 16-bit multiplexed address/data bus when enabled by jumpers on
CPU module.
ALE: CPU module's address latch enable signal
#RD: CPU module's READ signal
#WR: CPU module's WRITE (or WRITELOW) signal
#WRH: CPU module's WRITE (or WRITETHIGH) signal
#PSEN_A16: CPU module's PSEN signal (8051) or A16, where available
#CS0: CPU module's first chip select signal
#CS1: CPU module's second chip select signal
#CS2: CPU module's third chip select signal
#CS3: CPU module's fourth chip select signal
CAN0_RX/TX: Logic level connection to CPU module's first CAN module (where fitted).
USB-DEBUG+/−: USB signals connected to FTDI USB-JTAG device on CPU module
CNTRL_SPI_XX: Signals connected to CPU module's first SPI peripheral
CNTRL_I2C_SDA/SCL: Signals connected to CPU module's first or primary i2C channel. (This
is the i2C control backbone for the EDP baseboard).
CANH0/CANL0: CPU module's first CAN module via physical layer drivers.
VCC_CM: Peripheral operating voltage of CPU module currently fitted.
+3V3: +3V3 supply from baseboard voltage regulator
+5V: +5V supply from baseboard voltage regulator
SGND: Digital logic ground (connects to 12VGND at star point in baseboard)

1.4 The EDP Virtual CPU Concept

A microcontroller that has its I/O pins mapped appropriately onto the EDPCON1 and EDPCON2
connectors appears to be a virtual CPU to other I/O devices fitted on the bus. Thus for example, a 14-
bit ADC device on the EDPCON baseboard will see a CPU module also on the bus, as a virtual CPU
whose pinout is defined by the EDP bus. Currently two popular microcontrollers (Infineon XC167 and
ST STR9) have had their I/O pins mapped onto the EDPCON system. These two devices have some
features in common -UARTs, capture and compare pins, ADC, CAN but the STR9 also has USB
device. Thus the pin mapping to the EDPCON is not 100% in that on the XC167 version, the USB
device pins are unused. Both devices have dedicated motor control peripherals which although they
have different pin names, have virtually the same functionality.

Hence for example, a brushless DC motor control module with half-bridges can be designed to
interface to the motor control region of the EDPCON bus without any regard for the CPU type to be
ultimately used.
The net result is that subject some limitations, a range of modules bearing different CPUs can be
freely connected to a range of I/O modules.

The EDPCON has been designed to accommodate all the common peripherals found on current
microcontrollers, including advanced interfaces like SD/MMC and i2S. Thus it is possible to map
almost any microcontroller to this format.
1.4.1 Example of Real CPU to EDPCON Mapping

This is the mapping developed for the Infineon XC167 and used on the RS-EDP-CM-XC167 module.

1.4.1.1 Infineon XC167 – EDPCON1 Mapping

This mapping assigns the XC167 pins (and hence peripherals) into the appropriate regions on the EDPCON1 connector.
### 1.4.1.2 Infineon XC167 – EDPCON2 Mapping

<table>
<thead>
<tr>
<th>Pin</th>
<th>Function</th>
</tr>
</thead>
<tbody>
<tr>
<td>142</td>
<td>#RSTIN</td>
</tr>
<tr>
<td>3</td>
<td>#RSTOUT</td>
</tr>
<tr>
<td>23</td>
<td>SDA1</td>
</tr>
<tr>
<td>24</td>
<td>SCL1</td>
</tr>
<tr>
<td></td>
<td>Digital GND</td>
</tr>
<tr>
<td>116</td>
<td>AD15</td>
</tr>
<tr>
<td>115</td>
<td>AD14</td>
</tr>
<tr>
<td>114</td>
<td>AD13</td>
</tr>
<tr>
<td>113</td>
<td>AD12</td>
</tr>
<tr>
<td>112</td>
<td>AD11</td>
</tr>
<tr>
<td>111</td>
<td>AD10</td>
</tr>
<tr>
<td>106</td>
<td>AD9</td>
</tr>
<tr>
<td>105</td>
<td>AD8</td>
</tr>
<tr>
<td>102</td>
<td>AD7</td>
</tr>
<tr>
<td>101</td>
<td>AD6</td>
</tr>
<tr>
<td>100</td>
<td>AD5</td>
</tr>
<tr>
<td>99</td>
<td>AD4</td>
</tr>
<tr>
<td>98</td>
<td>AD3</td>
</tr>
<tr>
<td>97</td>
<td>AD2</td>
</tr>
<tr>
<td>96</td>
<td>AD1</td>
</tr>
<tr>
<td>95</td>
<td>AD0</td>
</tr>
<tr>
<td>93</td>
<td>#ALE</td>
</tr>
<tr>
<td>90</td>
<td>#RD</td>
</tr>
<tr>
<td>91</td>
<td>#WR</td>
</tr>
<tr>
<td>75</td>
<td>#WRH</td>
</tr>
<tr>
<td>16</td>
<td></td>
</tr>
<tr>
<td>7</td>
<td>CS0 (SRAM)</td>
</tr>
<tr>
<td>8</td>
<td>CS1 (CS8900)</td>
</tr>
<tr>
<td>9</td>
<td>CS2</td>
</tr>
<tr>
<td>84</td>
<td>CAN1 RX</td>
</tr>
<tr>
<td>87</td>
<td>CAN1 TX</td>
</tr>
<tr>
<td></td>
<td>USB DEBUG D+</td>
</tr>
<tr>
<td></td>
<td>USB DEBUG D-</td>
</tr>
<tr>
<td>76</td>
<td>SCLK0</td>
</tr>
<tr>
<td>67</td>
<td>MRST0</td>
</tr>
<tr>
<td>68</td>
<td>MTSR0</td>
</tr>
<tr>
<td>82</td>
<td>P4.2</td>
</tr>
<tr>
<td>25</td>
<td>SDA2</td>
</tr>
<tr>
<td>26</td>
<td>SCL2</td>
</tr>
<tr>
<td></td>
<td>NC</td>
</tr>
<tr>
<td></td>
<td>NC</td>
</tr>
<tr>
<td></td>
<td>NC</td>
</tr>
<tr>
<td></td>
<td>CANH</td>
</tr>
<tr>
<td></td>
<td>CANL</td>
</tr>
<tr>
<td></td>
<td>CPU’v Vcc 3V3 or 5V</td>
</tr>
<tr>
<td></td>
<td>Vcc 3V3 from reg</td>
</tr>
<tr>
<td></td>
<td>Vcc 5V from reg</td>
</tr>
<tr>
<td></td>
<td>Digital GND</td>
</tr>
</tbody>
</table>

*Notes: CAN0 and CAN1 (control physical layer CAN0/1)*
1.5 Inter-Module Communication

With up to four modules on the EDPCON bus, some form of inter-module communication is required. With a limited number of CPU pins available, it is necessary to use a serial communications protocol to, for example, take readings from a high-precision ADC at the same time as read a serial EEPROM on a further module. The I²C protocol is used as the main communication channel for such actions, although provision is made for SPI or even a CAN physical layer.

![Table of I²C Devices and Addresses](image)

<table>
<thead>
<tr>
<th>Module</th>
<th>I²C Device</th>
<th>Possible Range</th>
<th>Actual 7-bit Address</th>
<th>I²C channel</th>
<th>Actual 7-bit Address</th>
<th>I²C channel</th>
<th>Comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>Baseboard</td>
<td>PCF8575TS</td>
<td>0x20 - 0x27</td>
<td>0x20</td>
<td>CNTRL</td>
<td>XXXX</td>
<td>XXXX</td>
<td>DIP configuration switches</td>
</tr>
<tr>
<td></td>
<td>24C32 (Rev B)</td>
<td>0x50 - 0x57</td>
<td>0x51</td>
<td>CNTRL</td>
<td>0x52</td>
<td>CNTRL</td>
<td>4 kbyte EEPROM</td>
</tr>
<tr>
<td>Analogue AM</td>
<td>MAX1138EE</td>
<td>0x35</td>
<td>0x35</td>
<td>CNTRL</td>
<td>0x35</td>
<td>Gen0</td>
<td>12-channel 10-bit ADC</td>
</tr>
<tr>
<td></td>
<td>ADS263BRU50</td>
<td>0x2C - 0x2F</td>
<td>0x2C</td>
<td>CNTRL</td>
<td>0x2C</td>
<td>Gen0</td>
<td>Digital Potentiometer</td>
</tr>
<tr>
<td>Comms AM</td>
<td>PCF8583</td>
<td>0x50 - 0x51</td>
<td>0x50</td>
<td>CNTRL</td>
<td>XXXX</td>
<td>XXXX</td>
<td>Real-Time Clock and RAM</td>
</tr>
<tr>
<td>Digital AM</td>
<td>PCF8575TS</td>
<td>0x20 - 0x27</td>
<td>0x22</td>
<td>CNTRL</td>
<td>0x24</td>
<td>CNTRL</td>
<td>16-channel digital input</td>
</tr>
<tr>
<td></td>
<td>PCF8575TS</td>
<td>0x20 - 0x27</td>
<td>0x23</td>
<td>CNTRL</td>
<td>0x25</td>
<td>CNTRL</td>
<td>16-channel digital output</td>
</tr>
</tbody>
</table>

There are three possible I²C channels available although in most cases the default one (I2C_CTRL) will be sufficient. EDP modules that carry I²C device do, where possible, allow the user to configure the I²C addresses. This allows for example, up to three digital I/O modules to be fitted, with the GPIO devices on each module given an unique address. Where the address space of a particular I²C channel becomes full, devices can be connected to an alternative channel to get access to a completely new address space.

1.6 Inter-EDP System Communications

In a situation where there are multiple EDP baseboards, each with their own CPU modules in a complete system, an I²C bus can still be used to allow the CPUs to communicate, but the use of a CAN bus is strongly recommended. EDP I/O signals that are intended to be taken off-board are brought out on a standard DIN41612 64-way connector.
2. Using the EDP Baseboard

This section gives information on the features of the EDP baseboard, its connectors and the overall structure of the EDP system.

2.1 EDP Connectors

The EDP bus contained in the EDP baseboard is accessed through two Tyco-AMP 0.8mm pitch connectors. The signal names are intended to convey something of the capabilities of that signal. For example signal EVG0_GPIO40 is a pin that can generate timed events (i.e. pulses and pulse trains) as well as performing simple on/off pin control.
2.2 EDP Baseboard User Options Placement

There are a number of user-selectable functions on the baseboard, as shown below:
2.3 EDP Baseboard Component Placement

The location of the major items on the EDP baseboard is shown below.
2.4 EDP I/O Pin Headers

All the signals in the EDP backplane are available here on 0.1" pin headers for connection to test equipment, etc.

![EDP I/O Pin Headers](image)

<table>
<thead>
<tr>
<th>P001</th>
<th>P002</th>
<th>P003</th>
<th>P004</th>
<th>P005</th>
</tr>
</thead>
<tbody>
<tr>
<td>A01</td>
<td>A02</td>
<td>A03</td>
<td>A04</td>
<td>A05</td>
</tr>
<tr>
<td>A06</td>
<td>A07</td>
<td>A08</td>
<td>A09</td>
<td>A10</td>
</tr>
<tr>
<td>A11</td>
<td>A12</td>
<td>A13</td>
<td>A14</td>
<td>A15</td>
</tr>
<tr>
<td>A16</td>
<td>A17</td>
<td>A18</td>
<td>A19</td>
<td>A20</td>
</tr>
<tr>
<td>A21</td>
<td>A22</td>
<td>A23</td>
<td>A24</td>
<td>A25</td>
</tr>
<tr>
<td>A26</td>
<td>A27</td>
<td>A28</td>
<td>A29</td>
<td>A30</td>
</tr>
<tr>
<td>A31</td>
<td>A32</td>
<td>A33</td>
<td>A34</td>
<td>A35</td>
</tr>
<tr>
<td>A36</td>
<td>A37</td>
<td>A38</td>
<td>A39</td>
<td>A40</td>
</tr>
<tr>
<td>A41</td>
<td>A42</td>
<td>A43</td>
<td>A44</td>
<td>A45</td>
</tr>
<tr>
<td>A46</td>
<td>A47</td>
<td>A48</td>
<td>A49</td>
<td>A50</td>
</tr>
</tbody>
</table>

![EDP I/O Pin Headers Diagram](image)
2.5 Grounding Arrangements

The system ground (SGND) and 12V ground (12V_GND) are connected together at a star point on the baseboard. The 12V GND is used for high current devices like the motor controller and the Darlington output drivers on the digital I/O module. System ground is used for all returns on logic devices on all modules. It can be used for analogue returns but there is a risk of noise (ground bounce).

Analogue ground (VAGnd) by default is an offshoot of the system ground which occurs only on the CM. It is routed to the VAGnd pins of the CPU and also acts as a return for filter circuits used for analogue inputs. It is optionally possible to connect the SGND to the Analogue ground on the analogue module, although this should not be necessary unless there are a large number of resistive sensors being used. In this case, the link connecting VAGnd and SGND on the CM must be opened to avoid ground loops.

2.6 Positive Supplies

The +12V line comes via the screw terminals on the baseboard or the mini-jack. It is fused and filtered before entering the EDP backplane. The +3V3 and +5V voltage regulators are driven from the +12V.

2.6.1 Logic Supplies

Both +3V3 and +5V are available on the EDPCON to support both +5V and +3V3 processors and devices. To allow the interfacing of I/O devices at the required voltage, the positive supply to the CPU I/O domain is routed into the EDPCON through Vcc_CM. It is intended to be used for pull-ups on I/O pins and powering small active components that connect directly to the CPU such as discrete logic, op-amps etc. Vcc_CM is limited to 500mA total current draw from other modules and the baseboard.

Vcc_CM is connected inside the CM to the voltage used by the CPU’s I/O domain.
2.6.2 Analogue Supply

The Analogue supply to the CPU ADC may be derived from the local Vcc or from a precision reference located on the Analogue AM. Ideally the Analogue AM and CM should be in adjacent positions on the baseboard to keep the signal length to a minimum if the latter is chosen. The I²C ADC on the analogue module can use the Vcc_CM or the local precision voltage references, either +3V3 or +5V. The 5V reference is driven from the 12V to guarantee no drop-out problems.

As the anti-aliasing filters are run at +5V, the local ADC is not tied to the same voltage range as the CPU's ADC. It is the user's responsibility to make sure that the input does not exceed the permissible input voltage range of the CPU ADC. Protection resistors are provided to prevent damage.

2.7 Limits and Restrictions

Vcc CM max current .......................... 500mA
3V3 max current ................................. 2000mA
5V max current ................................. 2000mA
3V3 current + 5V current + Vcc_CM .......... 2000mA
SGND max current .......................... 2000mA
12V/GND max current ........................ 2000mA

Warning: Only attempt to fit two CPU modules to the baseboard at the same time when you are really certain you know what you’re doing! If they have different peripheral supply voltages then damage is likely to occur.
2.8 EDP Control Busses

2.8.1 I²C Busses

The EDP uses I²C as the data and control backbone. Depending on the capabilities of the CM fitted, up to three independent I²C busses are available. I²C channel “CNTRL_I2C” is the primary I²C device bus and is used by default to communicate with I²C devices on the baseboard and application modules.

The I²C address space is based on the 7-bit addressing scheme. I²C devices that are able to generate an interrupt request by default use the IRQ_GPIO16_CNTRL_I2C_INT line, with the option of using up to another three interrupt-capable lines. A pull-up resistor is provided on IRQ_GPIO16_CNTRL_I2C_INT so that the open collector /INT outputs on I²C devices can signal an interrupt by pulling this line down.

The I²C bus runs at +3V3 so any +5V devices must be connected via a level shifting mechanism.

The I²C bus devices require pull-up resistors on the SDA and SCL lines and these are incorporated on the baseboard.

There are three possible I²C channels available although in most cases the default one (I2C_CTRL) will be sufficient. EDP modules that carry I²C device do, where possible, allow the user to configure the I²C addresses. This allows for example, up to three digital I/O modules to be fitted, with the GPIO devices on each module given a unique address. Where the address space of a particular I²C channel becomes full, devices can be connected to an alternative channel to get access to a completely new address space.
2.8.1.1 Available $I^2$C Interrupt Request Lines

Each of the three potential $I^2$C channels has a dedicated interrupt request line into the CM. A spare interrupt line is provided that can be allocated to any channel, as defined by the user. However it is up to the user to make sure that the software is able to recognise the $I^2$C device that requested the interrupt.

<table>
<thead>
<tr>
<th>$I^2$C Channel</th>
<th>IRQ GPIO16_CNTRL_I2C</th>
<th>(integral pull-ups)</th>
</tr>
</thead>
<tbody>
<tr>
<td>$I^2$C_CTRL</td>
<td>IRQ GPIO18_GEN0_I2C</td>
<td>(integral pull-ups)</td>
</tr>
<tr>
<td>$I^2$C_GEN0</td>
<td>IRQ GPIO22_GEN1_I2C</td>
<td>(integral pull-ups)</td>
</tr>
<tr>
<td>$I^2$C_GEN1</td>
<td>IRQ GPIO24_I2C_INT</td>
<td>(integral pull-ups)</td>
</tr>
</tbody>
</table>

Uncommitted IRQ GPIO16_I2C_INT (integral pull-ups)
2.9 CAN Network

The on-board CAN network “CAN CNTRL” is intended to allow the interconnection of modules and other EDP systems via CAN. The first CAN module on any CPU is by default allocated to the CANH0 and CANL0 bus. This is the CAN physical layer (i.e. after the CAN transceivers) and can run at up to 1MB/s. The 120R termination resistors at the ends of the network are located on the CM and at the end of the baseboard that carries the Ethernet and USB connectors. If the CAN CNTRL bus is taken off-board via the DIN41612 expansion connector then the 120R resistor on the baseboard must be disconnected by removing the P201 link.

The CAN CNTRL bus is available through a 9-way D-connector on the optional EDP-AM-CO1-A communications module.

![Diagram of CAN network](image)

- Only CMs have 120R resistor
- Make solder bridge when CAN CNTRL is only used on baseboard. Default closed