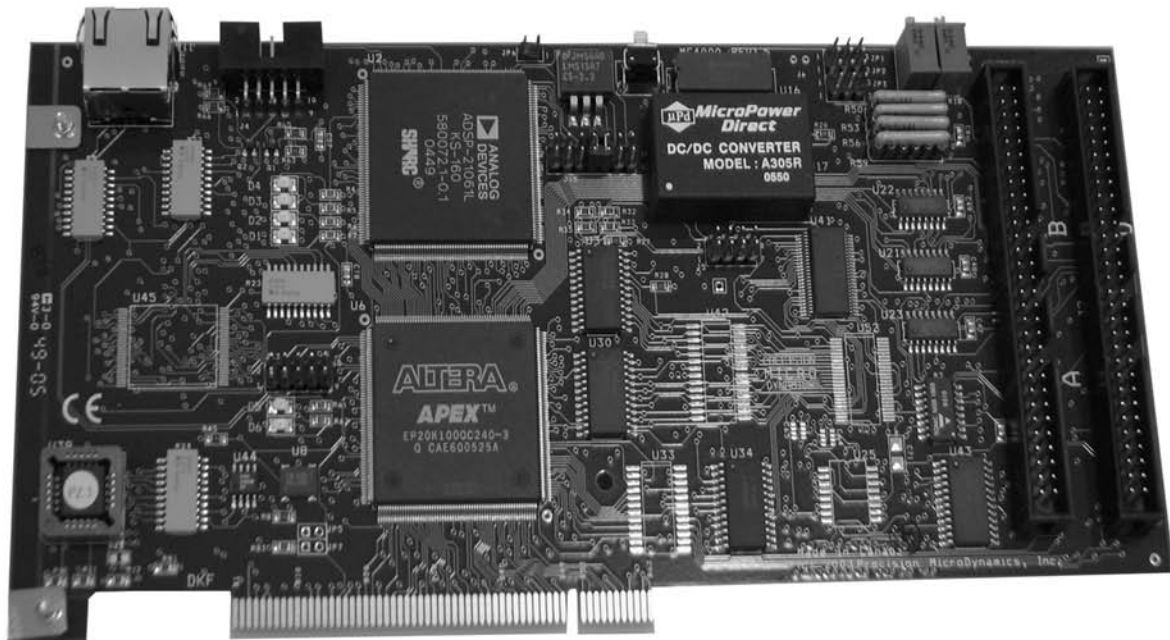


PHC Series

High Performance PC Based Controllers

- *Multi-axis, PCI bus motion control cards*
- *Suitable for brush servo motors, brushless servo motors, open and closed loop stepper motors*
- *Multi-axis synchronized motion*
- *Using standard suite of motion control software libraries and tools which provide a wealth of features*
- *Support MotionWire distributed network by adding extra digital I/O, servo/stepper modules easily*



PHC91 Series

- Buffered data acquisition
- Error mapping for high precision applications
- SIN/COS interpolation for sub-micron and sub-nanometer position resolution and control
- High speed position capture input and position compare output
- Low-pass and notch filters to reduce noise and eliminate unwanted mechanical resonances

Datasheet

| Specifications \ Model | PHC91-4A | PHC91-4A4S | PHC91-8A |
|--------------------------------|----------------------------------|----------------------------------|----------------------------------|
| Number of axis | 4 Servo | 4 Servo + 4 Stepper | 8 Servo |
| Quadrature decoders | 4 (10-60MHz) | 4 (10-60MHz) | 8 (10-60MHz) |
| Analog outputs | 4 (16-bit, ±10V) | 8 (16-bit, ±10V) | 8 (16-bit, ±10V) |
| Analog inputs | - | 4 (14-bit, ±10V) | - |
| Digital outputs | 8 | 8 | 16 |
| Digital inputs | 24 | 24 | 48 |
| Pulse & Direction outputs | - | 4 (2MHz) | - |
| Position capture inputs | - | 4 | - |
| Position compare outputs | - | 4 | - |
| Communication / Expansion port | PCI, MotionWire | PCI, MotionWire | PCI, MotionWire |
| Sampling rate, kHz | 4 | 4 | 4 |
| Programming languages | C/C++, VB, MotionTools Script | C/C++, VB, MotionTools Script | C/C++, VB, MotionTools Script |

Sample Program

```
...
...
...
int boardNumber          = 0;
int number_of_axis_to_move = 2;
int axis[2]              = {0, 1};
int dirn[2]              = {-1, -1};
int homeFlags[2]         = {0, 0};
double dest[2];
int ret;

ret = mciHomeToIndex(boardNumber, number_of_axis_to_move, axis, dirn, homeFlags);
ret = mciHomeDoneWaitAll(boardNumber, number_of_axis_to_move, axis, 10000);

while(1) {
    dest[0] = 50.0;    // move axis 0 to 50.0mm
    dest[1] = 100.0;  // move axis 1 to 100.0mm
    ret = mciVectorMoveAbs(boardNumber, number_of_axis_to_move, dest, axis);
    ret = mciMotionDoneWaitAll(boardNumber, number_of_axis_to_move, axis, 10000);

    dest[0] = 0.0;    // move axis 0 to 0.0mm
    dest[1] = 0.0;    // move axis 1 to 0.0mm
    ret = mciVectorMoveAbs(boardNumber, number_of_axis_to_move, dest, axis);
    ret = mciMotionDoneWaitAll(boardNumber, number_of_axis_to_move, axis, 10000);
}
...
...
...
```