

QMotor 3.0

A PC Based Real-Time Multitasking
Graphical Control Environment

The screenshot displays the QMotor 3.0 software interface with several windows open:

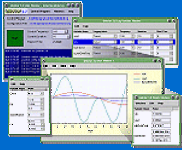
- QMotor 3.0 Main Window - InductionMotor.cp**: Shows the control program configuration. The "Start" button is highlighted in green. The control frequency is set to 100, and the control duration is 10. The elapsed time is 2.4 sec. A log window shows the following messages:

```
Jun 8 14:45: Welcome to QMotor 3.0
Jun 8 14:45: Loaded control program InductionMotor.cp
Jun 8 14:45: Loaded configuration file InductionMotor.cfg
Jun 8 14:45: Control Program
Jun 8 14:45: Control Program
Jun 8 14:45: Control Program
Jun 8 14:45: Control Program
```
- QMotor 3.0 Log Variable Window**: A table showing the logging configuration for variables:

Variable Name	Logging Mode	Frequency	Duration	Start Time
vout	Timed	100	10	0
vin	Timed	100	10	0
logVector	Timed	100	10	0
d_vinOverTwo	Timed	100	10	0
- QMotor 3.0 Plot Window 1**: A graph showing the response of the control system over time (0 to 10). The plot displays three variables: v_{in} (red line), v_{out} (green line), and $d_{vinOverTwo}$ (blue line). The v_{out} signal shows a damped oscillation, while v_{in} and $d_{vinOverTwo}$ show smoother curves.
- 3.0 Control Parameter**: A window showing the control parameters for the system:

Variable Name	Value
Offset	0.53
Kp	-3.3
ScaleFactor	5
Kd	0.211
	0.002
	0.232
- QMotor 3.0 Watch Window**: A window showing the current values of the variables being monitored:

Variable Name	Value
vin	1.33366
vout	3.95841
vinOverTwo	0.678867
logVector	2.68503e-294
	3.05375e-270
	3.1975e-313

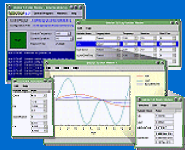


QMotor 3.0

A PC Based Graphical Control Environment

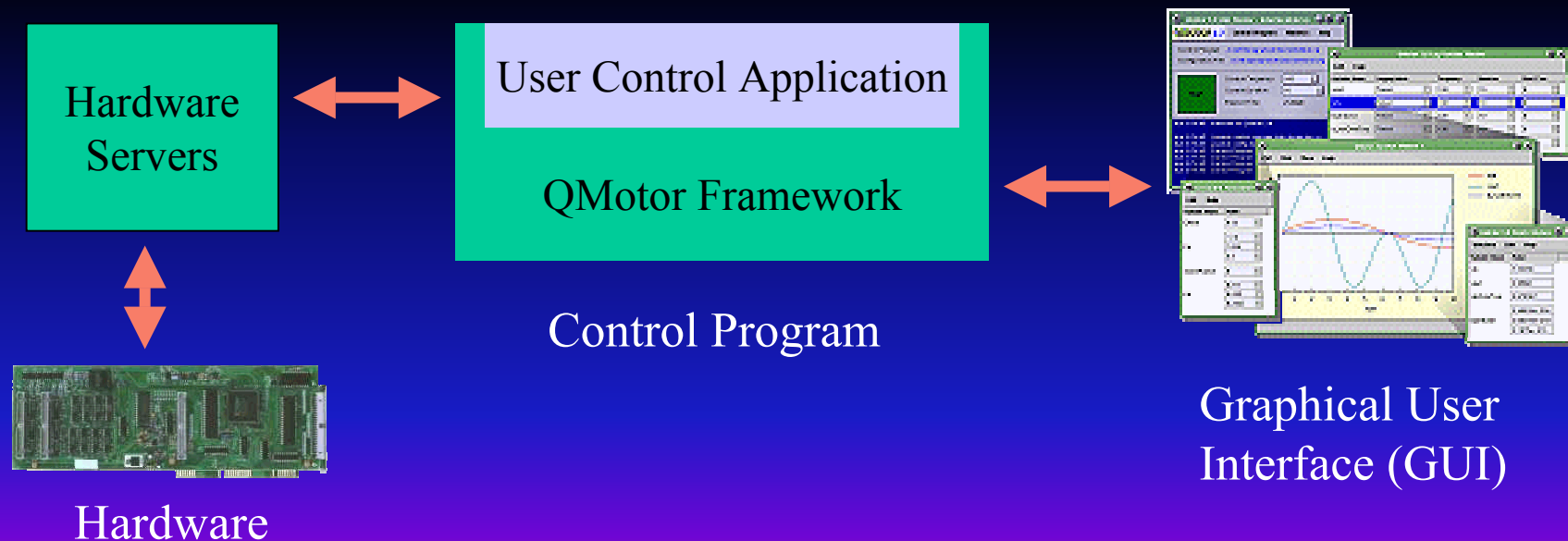
Outline

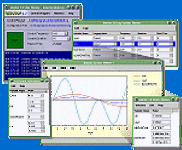
- Introduction
- Advantages Of The QMotor Design
- Client/Server Architecture
- The QMotor Development Cycle
- Live Demonstration



QMOTOR 3.0 is an environment for control development.
It consists of three parts, which run as separate programs:

- The *Control Program* (using the QMOTOR C++ framework) implements the user control application
- The *Hardware Servers* to access hardware components
- The *Graphical User Interface (GUI)* for control testing and tuning





QMOTOR 3.0

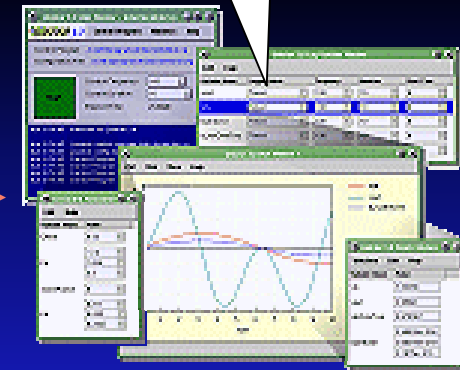
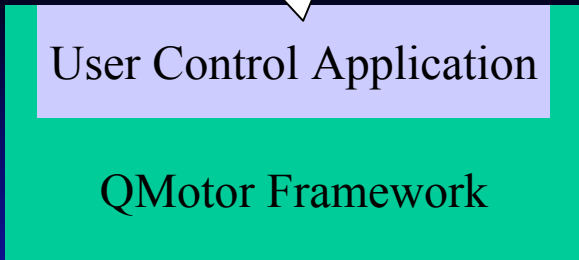
A PC Based Graphical Control Environment

Advantages of the QMotor 3.0 Design

Hardware Sharing: Multiple Control Programs can share the hardware of a board

Security: The control program doesn't need privileged access to the systems hardware.
Standalone Mode: Running control programs without the GUI

Timing: The GUI can not delay the control program as it runs at a lower priority

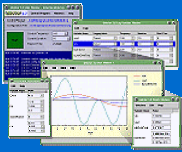


Hardware

Control Program

Graphical User Interface (GUI)



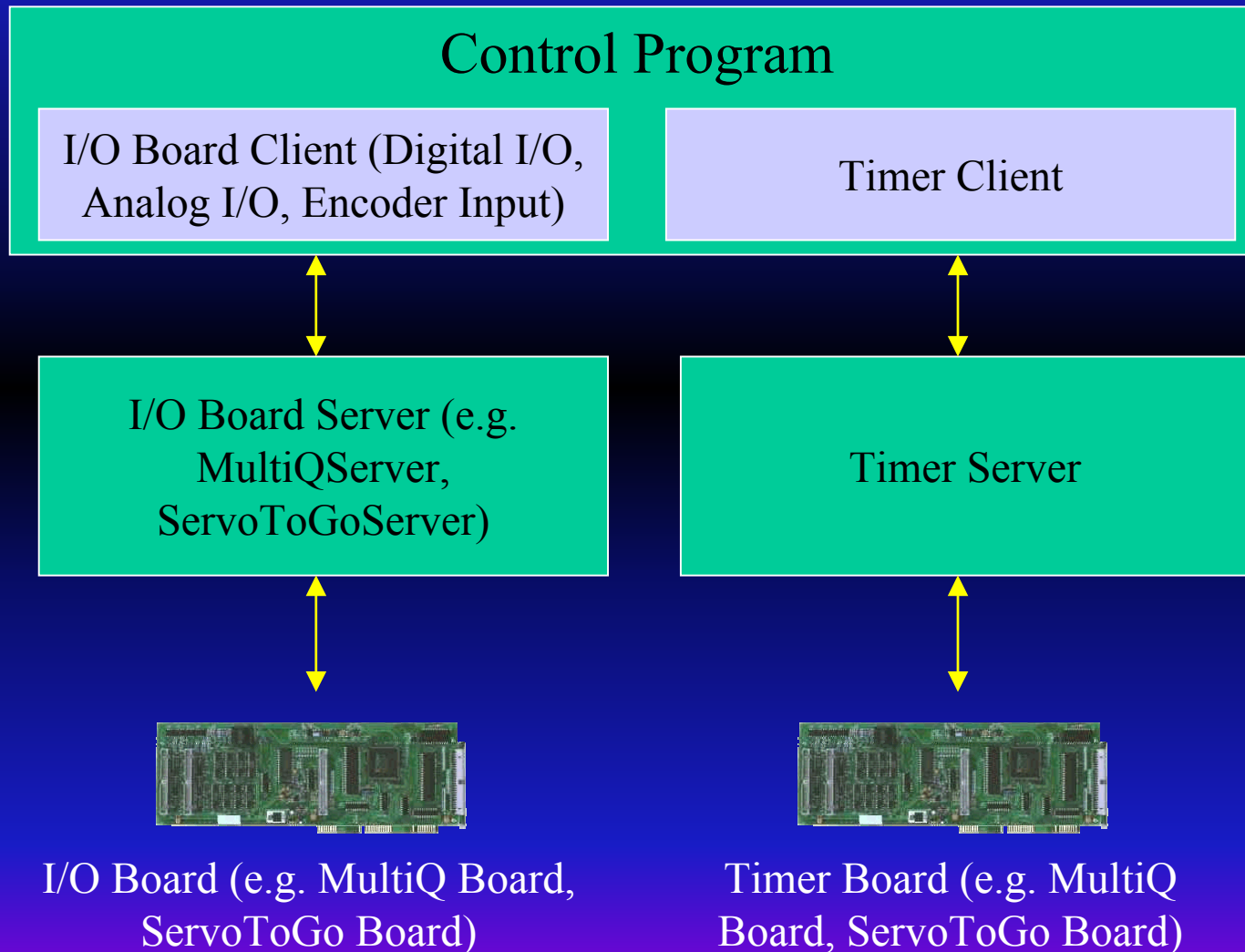
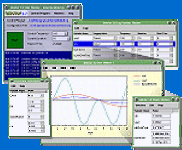


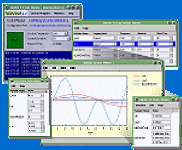
Maximum Flexibility

- QMotor is not limited to specific hardware
- The control program can run standalone
- Extensive logging options

Pure PC Based Solution

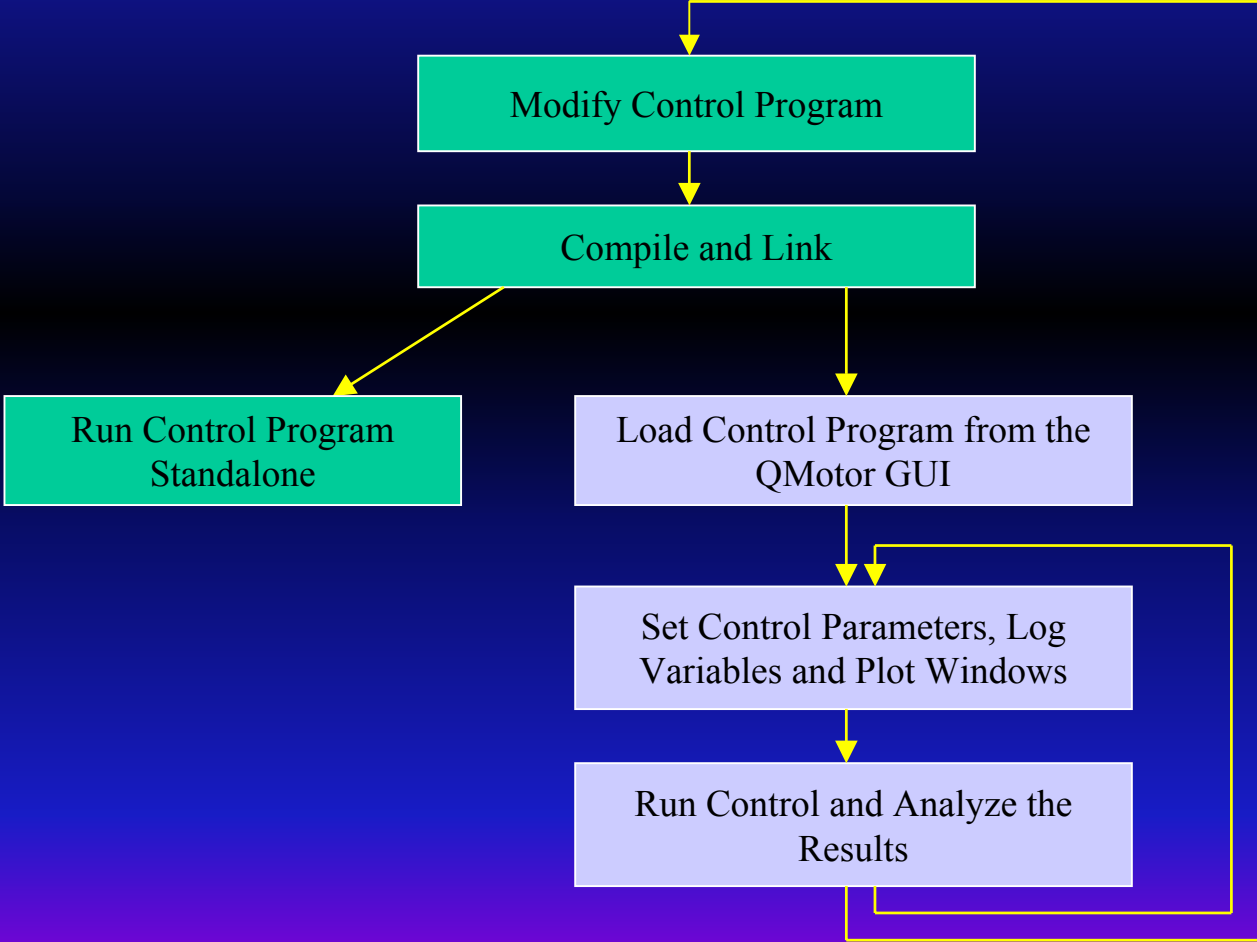
- No need for DSP boards
- Low hardware costs (PC and hardware boards)
- The PC is a technically advancing technology (speed, availability of hardware and software components)
- Runs on QNX, a high quality real-time operating system

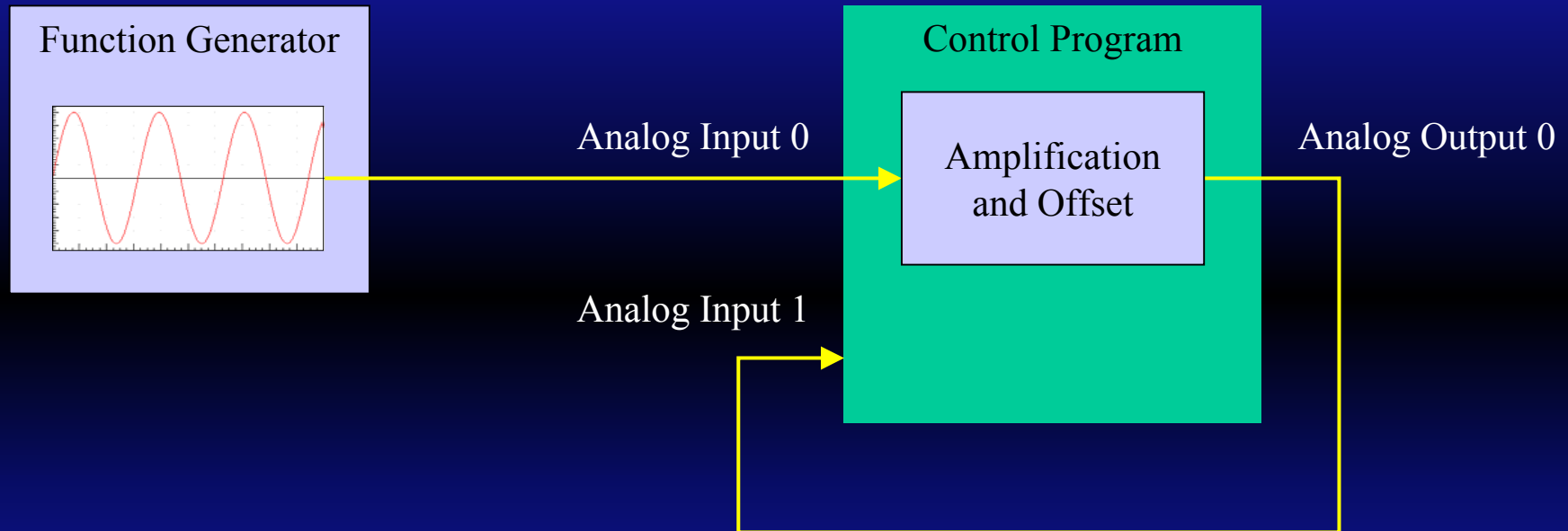


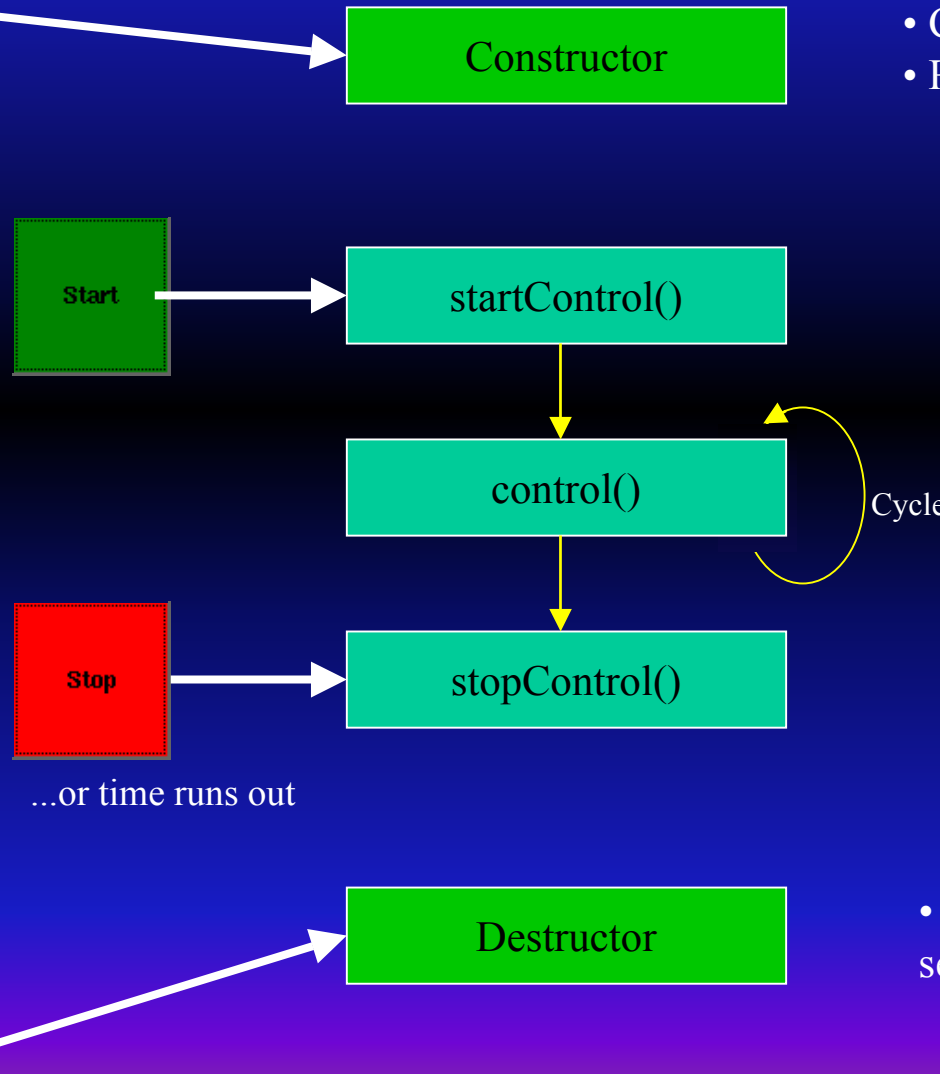
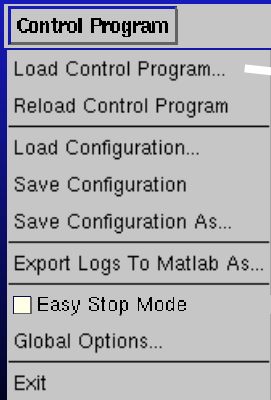


What are the steps in developing control programs with QMotor?

Create a new control program from the template







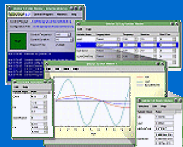
- Connect to the hardware servers
- Register control and log variables

- Initialize variables

- Hardware input
- Control calculations
- Hardware output

- Zero analog outputs

- Disconnect from the hardware servers



QMOTOR 3.0

A PC Based Graphical Control Environment

More Information...

Quality Real-Time Systems

<http://qrts.com>

info@qrts.com

Online Manual

<http://qrts.com/products/qmotor/manual/main.html>

QNX Helpviewer