

Contents	Page
1. MACHINE DESCRIPTION	1
2. REQUIREMENTS	1
3. COMPONENTS SELECTED	1
4. IMPLEMENTATION	2

### 1. Machine Description

This textile application is a computer-controlled quilt stitcher. After a quilt has been pieced together to its final size with a standard sewing machine, it must be stitched to a batting material and a backing material. A computer-controlled stitcher is capable of creating intricate XY patterns on the quilt, and these patterns are drawn in a CAD program beforehand.

The stitcher consists of a two-axis gantry mechanism that moves the needle in the XY plane. A third axis drives the needle, which is driven in and out of the stationary quilt material. The needle motion is synchronized with the XY motion such that the number of stitches per inch is a constant. A mechanical cam links the Z axis motor to the needle, such that if the Z motor spins at a constant speed, the needle reciprocates at a constant rate.

The machine contains a PC, which allows the user to select stitch patterns from a user interface and even create new patterns.

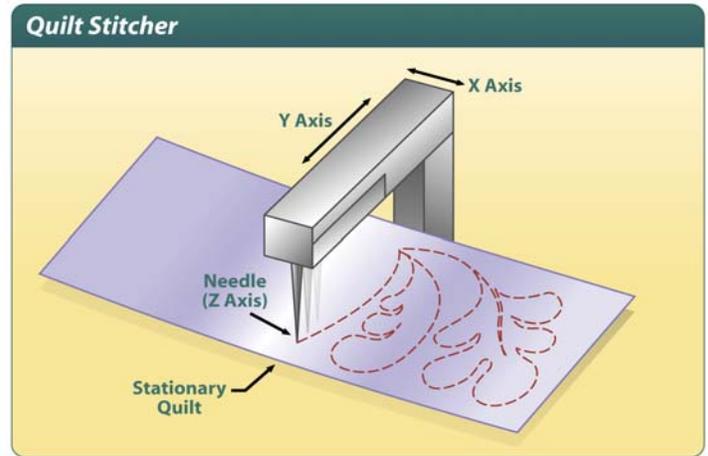


Figure 1.

### 2. Requirements

This section summarizes the requirements for the machine described above:

- (1) Three axes (XYZ)
- (2) Forward and reverse limit switches for X and Y
- (3) Manual homing
- (4) XY coordinated to follow any two-dimensional path specified by CAD
- (5) Constant stitch length (regardless of speed)
- (6) Graphical User Interface
  - User selects from a list of patterns
  - Selected pattern is displayed
  - Start button downloads path to controller and begins path

### 3. Components Selected

This section describes the Galil hardware and software products chosen to implement the machine's control system. Below is a complete bill of materials followed by a description of major components.

Table 1. Bill of Materials for Quilt Stitcher Control System

Part Number	Description	Unit Price (U.S.) Qty 1 / qty 100
DMC-1832	PCI Bus 3-axis motion controller	\$1045/ \$725
CABLE-100-1m	High density cable in 1 meter length	\$125
AMP-19540	4-axis amplifier for driving four brush or brushless servos	\$795/ \$495
BLM-N23-50-1000 or equivalent	NEMA 23 Brushless Servo Motor, 1000 ppr encoder X3	Consult mfg.
CPS-12-24 or equivalent	Power Supply 12A, 200 Watt	Consult mfg.
WSDK Servo Tuning Software	Servo Tuning and Analysis Software	\$195 (one time)
CADtoDMC	DXF to DMC translator	\$595 (one time)
ActiveX Tool Kit	Visual Basic/ActiveX tools in 16- and 32-bit	\$595 (one time)
Visual Basic 6.0	Visual Basic development environment from Microsoft	Contact Microsoft

## Controller: DMC-1832

Since this application requires a host PC, a bus-based controller is appropriate. Since we only have three axes and little I/O, the PCI-based DMC-1832 is appropriate. The PCI bus allows for rapid transfer of the path data from the PC to the controller.



Figure 2. DMC-1832

## Motor: BLM-N23-50-1000

For maintenance-free operation, we choose brushless motors. Galil's NEMA 23 #BLM-N23-50-1000 brushless motors, or equivalent, are appropriate because all axes require less than 0.3 Nm of continuous torque. Incremental encoders with 1000 cycles per revolution are installed on the motors resulting in 4000 quadrature counts per revolution. Hall sensors are not required on the motors as the incremental encoders provide commutation tracks for input to the amplifiers.

## Amplifier: AMP-19540

To drive the three motors, we choose the very compact AMP-19540, which is a four-axis brushless amplifier (500 W per axis).



Figure 3. AMP-19540

## CADtoDMC

CADtoDMC is Galil software for Windows that takes an industry standard .DXF CAD file and converts it into a Galil language .DMC file. This software can either be used as a stand alone program with its own GUI, or its functionality can be embedded into your own program (VB, C...) and GUI. The program simply prompts the user for a DXF file (which must conform to certain rules described in the CADtoDMC manual) and then displays the motion path when it is done converting. A .sen file will be generated in the same directory as the .dxf file. A .sen file is a Galil language file created by CADtoDMC which is sent to and executed by the controller line by line, where-

as the .dmc file is made to be downloaded to the controller memory and then executed.

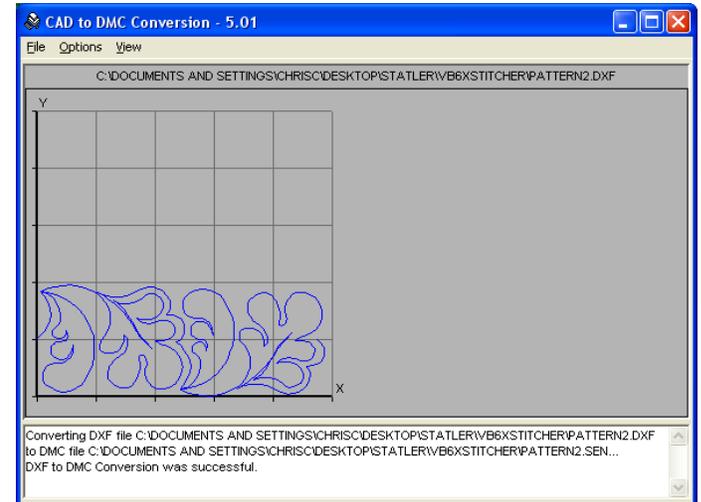


Figure 4. Galil's CADtoDMC software

## ActiveX Toolkit

The ActiveX Toolkit provides ready-made GUI components that can be dragged and dropped into your application's GUI. The ActiveX Toolkit often greatly reduces software development when implementing a GUI that communicates with a Galil controller. For this application, we use the DMCMove object, which provides a picture of the stitch pattern.

## 4. Implementation

This section details how the components selected above were used to implement the control system.

### Vector (VM, VP)

The application requires that the XY axes be coordinated to follow a two-dimensional path. To meet this need, CADtoDMC uses the Galil controller's vector mode. Vector mode allows two axes to be linked together to perform linear and circular interpolation such that complex XY patterns can be executed. In vector mode, the X and Y axes are linked together and referred to as the S axis. Vector speeds anywhere along the path can be specified to tailor the motion profile (see VP command < and > operators in command reference). The simple example below draws a square of 1000 counts per side in vector mode:

```
VMXY           ;'specify xy axes for vector mode
VP1000,0       ;'specify points to travel through
VP1000,1000
VP0,1000
VP0,0
VE             ;'end the vector sequence
BGS           ;'begin motion
AMS           ;'wait until motion is complete
```

## Gearing (GA, GR)

The application requires that the stitch length be constant, regardless of the speed. To meet this need, we use an advanced gearing feature of the controller that links the motion of one axis (Z) to the vector motion of two axes (XY or S). The Z motion is proportional to the arc length along the XY path. This causes the stitch length to be constant regardless of how fast the XY axes traverse the stitch path. The simple example below gears the Z axis to the XY vector path length. The gear ratio depends on the desired stitch length and the drive train connecting the Z axis motor to the needle.

```
GAZ=S      ;'gear z axis to xy path length
GRZ=0.41   ;'set gear ratio
```

## Program Organization

The PC software is organized as a chain (figure 5). First, the pattern designer enters by hand a stitching pattern into a CAD software package (such as AutoCAD). She then saves the finished drawing as a DXF file. The pattern designer then runs CADtoDMC and selects the DXF file, which is converted into a Galil-language .sen file. The machine user finally runs the Visual Basic application and selects the pattern from a list and hits the Run button to begin stitching the pattern.



Figure 5. Control Software Block Diagram

## Graphical User Interface

When the Quilt Stitcher Visual Basic application is launched, it displays what controller it is connected to and presents the user with a list of stitch patterns. When the

user clicks on a stitch pattern file (left), a two-dimensional image of the pattern is displayed on the right (with the DMCMove ActiveX Toolkit object).

DMCMove is a component of the Galil ActiveX Toolkit that takes a Galil Language file as input (such as one generated by CADtoDMC) and produces an image of the path as output. This provides visual feedback so the user is sure she has selected the correct pattern before she runs it. When the user has selected the correct stitch pattern, she hits the Run button to initiate stitching.

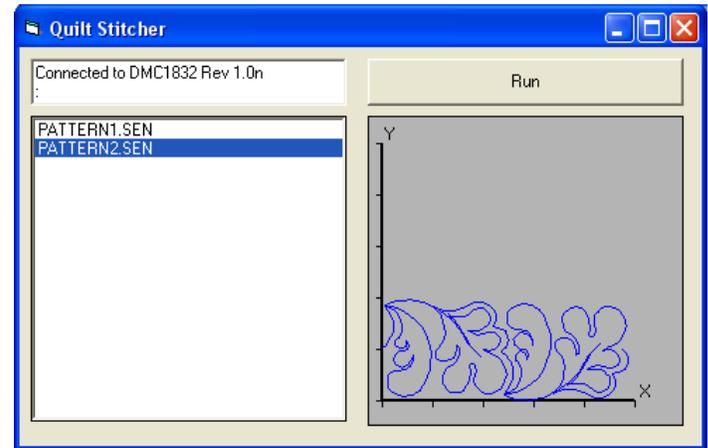


Figure 6. Quilt Stitcher Graphical User Interface

## Visual Basic Code

This section shows the visual basic code for the application. It consists of four routines that handle the following events:

- (1) Launching the program
- (2) Clicking on the file list
- (3) Clicking the Run button
- (4) Closing the program

The comments next page explain the code in detail.

Figure 7. Visual Basic Code for Quilt Stitcher GUI

```
Private Sub Form_Load()                                'runs when program starts
    DMCSHELL11.DMCConnect = True                       'connect to the Galil controller

    DMCSHELL11.DMCCommand = Chr$(18) + Chr$(22)       'send ctrl-R ctrl-V to query controller type
    Text2.Text = "Connected to " + DMCSHELL11.DMCResponse 'display controller type
    DMCSHELL11.DMCCommand = "MO"                       'turn motors off so user can home manually
End Sub

Private Sub File1_Click()                              'file selection box event procedure
    DMCMove1.DMCFileName = File1.FileName              'inform DMCMove of .sen file to display
    DMCMove1.Open                                     'display the file
End Sub

Private Sub Command1_Click()                           'run button event procedure
    If File1.FileName = "" Then                       'make sure the user has selected a file
        MsgBox ("Please select a file")                 'put up a message box if no file selected
    Else
        DMCSHELL11.DMCFileName = File1.FileName       'copy selected file name to ActiveX toolkit
        DMCSHELL11.DMCFileOperation = FileOperationDownload 'select file operation download to controller
        DMCSHELL11.DMCCommand = "GAZ=S"               'gear needle to vector path
        DMCSHELL11.DMCCommand = "GRZ=0.41"           'set needle gear ratio
        DMCSHELL11.DMCCommand = "SH"                 'turn motors on
        DMCSHELL11.DMCCommand = "XQ"                 'execute path
    End If
End Sub

Private Sub Form_Unload(Cancel As Integer)             'runs when program ends
    DMCSHELL11.DMCCommand = "ST"                       'stop axes and program running on controller
    DMCSHELL11.DMCConnect = False                     'terminate communication to controller
End Sub
```