

Dios 40 to Mark III Conversion

I received several requests about using the Dios on the MARK III. After some experiments I came up with the modifications needed to use the Dios on the MARK III controller board. The Mark III has a RS232 Driver which is good but its connected to the UART. The Dios does not use the UART for programming. What we need to do is rewire the MARK II board to connect the driver to port 16 and 17.

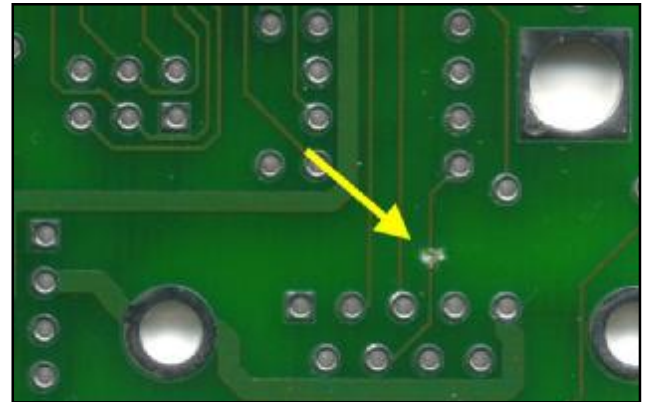
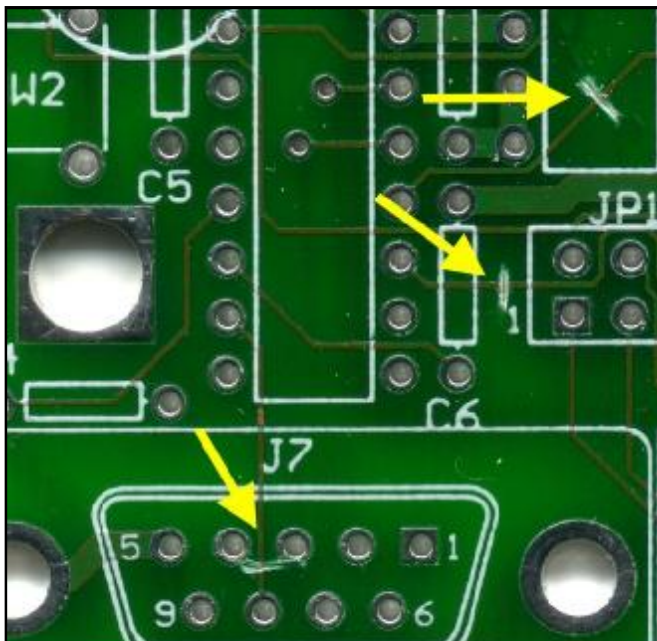
Warning these modifications are permanent and you can not go back to using the 16F877 after they are made. You will need the following.

- Dios Ultra 40 Pin Chip
- 10Mhz Resonator
- 4 1" pieces of 28ga wire

The conversion will cost you about \$30 and once done you can program your MARKIII with the power of the Dios. The PC software is free and can be downloaded from the Kronos Robotics web site

Step 1

First thing you need to do is cut 4 traces. Three on the top of the board and 1 on the bottom.



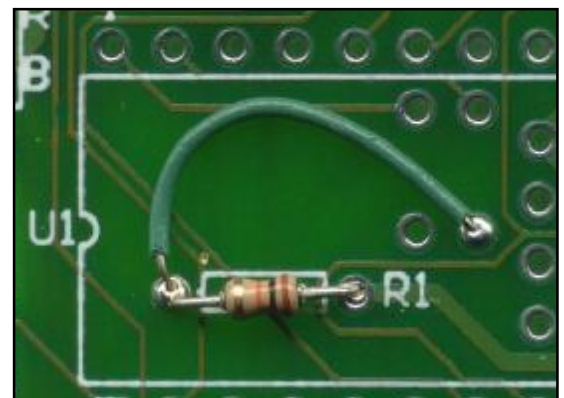
I used a small hobby knife to cut the traces. And while these mods are very easy to make to a new board you can with care make them to a board that has already been assembled. Just cut the indicated trace on J7 where it first emerges from the connector.

Step 2

Once the traces are cut you need to wire in 4 jumpers. These jumpers are as follows:

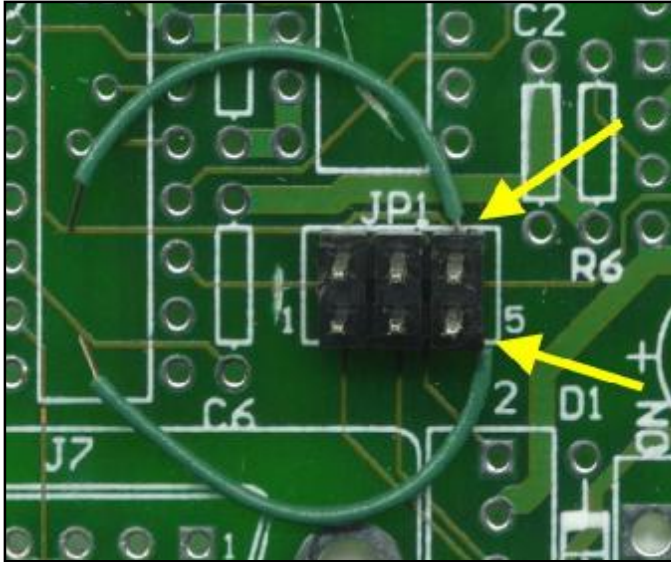
- DS232 pin 8 to DB9 pin 4
- DS232 pin 9 to Dios pin 1
- DS232 pin 11 to Dios pin 7
- DS232 pin 12 to Dios pin 6

You can make the connection under the board or with small 28ga wires into the holes of a new board. Just don't solder the wires until the original components is installed into the same hole.

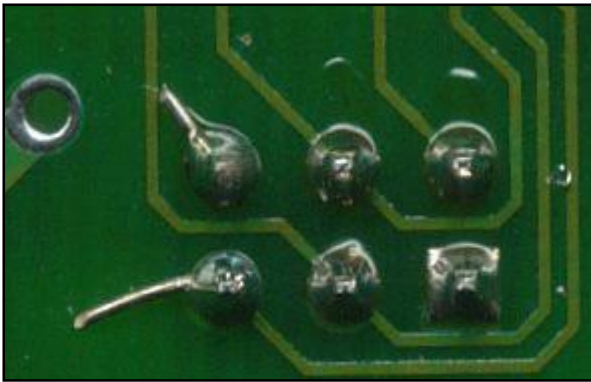


This jumper connects DS232 pin 9 to the ATN pin of the Dios.

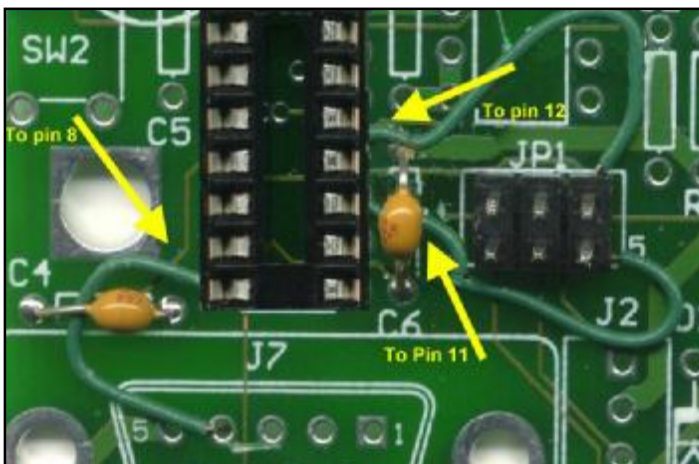
Dios - Mark III



These two connections connect to the Dios pins 6 and 7 (IO ports 17/16)



Here you can see the under side of the 6 pin connector with the small jumpers sticking out.



This shows the jumpers connected.

Step 3

The Dios Ultra uses a 10 Mhz Resonator. It uses a 4x PLL inside to achieve its 40 MHz clock. You must install a 10 Mhz resonator into position marked Y1. If you are converting a board that has already been built you will need to remove the old resonator. This can be done a couple of ways. The easiest tis to snip the leads and just solder the new resonator on top of the old leads.

Here is a completed MARK III controller board. Notice on this board I did not install the small 6 pin connector. I also installed 3 female headers for connecting the front sensors. (and other sensors)



Connections

- Left servo: Port 5 use setLservo(microseconds)
- Right Servo: Port 6 use setRservo(microseconds)
- Right Edge Sensor: Port 32 (A5) use AtoD(5)
- Middle Edge Sensor Port 31 (A6) use AtoD(6)
- Left Edge Sensor: Port 30 (A7) use AtoD(7)
- Left Range Sensor: Port 18 (A3) use AtoD(3)
- Right Range Sensor: Port 19 (A2) use AtoD(2)
- JP1 pin 1 (SCL): Port 12 ,hardware SCL
- JP1 pin 2 (SDA): Port 11, hardware SDA
- JP1 pin 5 (LSCL): Port 16, debug Xmt
- JP1 pint 6 (RSCL): Port 17, debug Rx

Software

This program turns the MARKIII into a small remote controlled bot. You hook up as Vishay IR Module and Use the channel and volume keys of a Sony Remote to control the bots direction. The MARKIIIinit sets up a High speed servo handler that's run in the background. You just need to make sure that you just off the IRQ's before doing any timing critical operations. We use the irqglobalstop and irqglobalstart commands to do this. This kind of program is good to use when testing sensors. Use the setRservo and setLservo commands to setup the timing sent to the servos.

```
MARKII test 1
func main()
gconst tservol 5
gconst tservor 6
output 4
print "Reset"

MARKIIIinit

dim cmd,x1,x2,x3,counter

'-----
'Used by IR Module
'Im providing power to IR Module with ports 0 and 1. 0=+ 1=- and 2 = input
output 0,1
high 0
low 1
'-----

setRservo(1500)
setLservo(1500)

botstop

dim runflag
runflag =0
loop:

irqglobalstop
cmd = IRread(2,5000)
irqglobalstart
if cmd = 0 then
inc counter
if counter > 3 and runflag = 1 then
counter = 0
print "stopit"
runflag = 0

botstop
endif

goto loop
endif

counter = 0
print cmd
irqglobalstart
if cmd = 17 then
botfwd
runflag = 1
goto loop
endif

if cmd = 18 then
botbwd
runflag = 1
```

```
goto loop
endif

if cmd = 19 then
bottright
runflag = 1
goto loop
endif

if cmd = 20 then
botleft
runflag = 1
goto loop
endif

goto loop

endfunc

'-----
'MARKIII stuff begins here
'-----

func botstop()
rstop
lstop
endfunc

func botleft()
rfwd
lbwd
endfunc

func botright()
lfwd
rbwd
endfunc

func botbwd()
lbwd
rbwd
endfunc

func botfwd()
lfwd
rfwd
endfunc

func rstop()
setRservo(1500)
endfunc
func lstop()
setLservo(1500)
endfunc

func rfwd()
setRservo(1000)
endfunc

func rbwd()
setRservo(2000)
endfunc

func lfwd()
setLservo(2000)
endfunc

func lbwd()
setLservo(1000)
endfunc

func setRservo(tspeed)
dim tnum as float
tnum = .716 * tspeed
SERVOR = tnum
endfunc
```

Dios - Mark III

```
func setLservo(tspeed)
dim trum as float
trum = .716 * tspeed
SERVOL = trum
endfunc

'-----
'Must be called first.
'Pass the number of servos to process 1-4
'If you pass 0 the routines shuts down
'The Servos are tied to ports 0-3
'-----

func MARKIIIinit()
global SERVOMAX
global SERVOMAXCOUNTER
global SERVOCOUNT
global SERVOL
global SERVOR
global SERVORESET
global SERVOCOUNT

SERVOCOUNT = 2

output 5,6
SERVORESET = 47500

startIRQ:
'Some IRQ Settings
irqTMR1start 'Start Timer IRQ
T1CON= 177
irqperipheralstart
irqglobalstart

SERVOMAX=2000

'Default servo pos
SERVOR = 1000
SERVOL = 1000

endfunc

'-----
'Main servo handler
'-----

startirqasm TMR1

bcf T1CON,0 ;Turn it off incase we spend too much time here

;-----
doservo1
bsf ioport tservol

movff G_SERVOLh,G_SERVOCOUNTRh
movff G_SERVOLI,G_SERVOCOUNTI

movff G_SERVOMAXh,G_SERVOMAXCOUNTERh
movff G_SERVOMAXI,G_SERVOMAXCOUNTERI
movlb .2
loopx1

movf G_SERVOCOUNTI,f
btfsc STATUS,Z
decf G_SERVOCOUNTRh,f
decf G_SERVOCOUNTI,f

tstfsz G_SERVOCOUNTI
goto domainpwm1
tstfsz G_SERVOCOUNTRh
goto domainpwm1
bcf ioport tservol

domainpwm1

movlb .2
```

```
movf G_SERVOMAXCOUNTERI,f
btfsc STATUS,Z
decf G_SERVOMAXCOUNTERh,f
decf G_SERVOMAXCOUNTERI,f

tstfsz G_SERVOMAXCOUNTERI
goto loopx1
tstfsz G_SERVOMAXCOUNTERh
goto loopx1

movlw .1
;subwf servocount,w
subwf G_SERVOCOUNTI,w
btfsc STATUS,Z
goto alldone

;-----
doservo2
bsf ioport tservor
movff G_SERVORh,G_SERVOCOUNTRh
movff G_SERVORI,G_SERVOCOUNTI

movff G_SERVOMAXh,G_SERVOMAXCOUNTERh
movff G_SERVOMAXI,G_SERVOMAXCOUNTERI

loopx2
movf G_SERVOCOUNTI,f
btfsc STATUS,Z
decf G_SERVOCOUNTRh,f
decf G_SERVOCOUNTI,f

tstfsz G_SERVOCOUNTI
goto domainpwm2
tstfsz G_SERVOCOUNTRh
goto domainpwm2
bcf ioport tservor

domainpwm2
movf G_SERVOMAXCOUNTERI,f
btfsc STATUS,Z
decf G_SERVOMAXCOUNTERh,f
decf G_SERVOMAXCOUNTERI,f

tstfsz G_SERVOMAXCOUNTERI
goto loopx2
tstfsz G_SERVOMAXCOUNTERh
goto loopx2

movlw .2
;subwf servocount,w
subwf G_SERVOCOUNTI,w
btfsc STATUS,Z
goto alldone

alldone

;Turn timer on and reset counter
bsf T1CON,0
movff G_SERVORESETh,TMR1H
movff G_SERVORESETI,TMR1L

movlb .0

endirqasm

include \lib\DiosIR.lib
```

Related Products

DiosPro 40	http://kronosrobotics.com/xcart/customer/product.php?productid=16428
EZRS232 Driver	http://kronosrobotics.com/xcart/customer/product.php?productid=16167
Dios Componets Package	http://kronosrobotics.com/xcart/customer/product.php?productid=16172
10Mhz Resonator	http://kronosrobotics.com/xcart/customer/product.php?productid=16139
Vishay IRModule	http://kronosrobotics.com/xcart/customer/product.php?productid=16226
9 Pin Cable	http://kronosrobotics.com/xcart/customer/product.php?productid=16259

