

Question's and Answers for Dios Class Microcontrollers

How do I create 6 50Hz PWM signals with the Dios?	1
What kind of cable do I need to connect to the Dios carrier boards?	2
Can the Dios clock frequency be changed?	2
I cant seem to get the low power when I put the Dios Chip to sleep. Any hints?	3
How much RAM is available on the Dios?	3
Can I add more memory to the Dios?	3
What is the maximum voltage the AtoD ports can take?	3
What is the conversion time of the AtoD on the Dios?	4
How do I use global constants in my inline assembly code?	4
What circuit does the POT and Rctime expect?	4
If I call a subroutine with the gosub command can I exit a function while in that routine?	5
Are there any increment and decrement short cuts?	5
Can I create my own default template for use when I create a new Dios file?	6
Can the IO ports be pulled up or down internally?	6
Is there a way to clear the serial UART?	6
Can I create my own help files for my libraries?	6
Is there a fast way to load and edit a include file?	6
Can I drop files onto the Dios editor form?	6
Is there a line continuation character available on the Dios Compiler?	6
Can I run more then one instance of the Dios Compiler?	7
Is there pre/post increment/decrement math in dios language like in c?	7
Is there a way to see how many bytes there are waiting in the hserin buffer?	7
How do I determine how much variable space is available on the fly?	8
Can the resolution of the pulsein command be changed?	9
What determines the state of the puleout command?	9
If a hardware IRQ is generated while the my Dios program is using the pause command what happens.	9
Can I use the onirq command with timer1,timer2, and timer3?	10
What does FOSC/4 mean?	10
Does the Dios Language support parentheses in math expressions?	10
What floating point format does the Dios use?	10
Does the Dios support recursion? If so how deep?	11
Is there a way of converting a string valu to an integer value similar to the val function in VB?	11
How do I shift bits right or left with the Dios language?	11
Does the Dios Language Support signed numbers in its variables?	11

How do I create 6 50Hz PWM signals with the Dios?

Its just a matter of setting up some counters and turning the ports off when a certain count is reached.

```
'Multi 50hz pwm signals
func main()
```

```
output 1,2,3,4,5,6
low 1,2,3,4,5,6
```

```
dim maincounter
dim off1,off2,off3,off4,off5,off6
```

```
'set your duty cycles here
```

```
off1 = 50
```

```
off2 = 50
```

```
off3 = 50
```

```
off4 = 50
```

```
off5 = 50
```

```
off6 = 50
```

```
mainloop:
```

```
high 1,2,3,4,5,6
```

```
pauseus 1025 'Tune the frequency here
```

```
for maincounter = 1 to 100
```

```
  if maincounter = off1 then
```

```
    low 1
```

```
  endif
```

```
  if maincounter = off2 then
```

```
    low 2
```

```
  endif
```

```
  if maincounter = off3 then
```

```
    low 3
```

```
  endif
```

```
  if maincounter = off4 then
```

```
    low 4
```

```
  endif
```

```
  if maincounter = off5 then
```

```
    low 5
```

```
  endif
```

```
  if maincounter = off6 then
```

```
    low 6
```

```
  endif
```

```
next
```

```
goto mainloop
```

```
endfunc
```

The pause command in the main loop is used to set the frequency. There other ways of doing this for faster frequencies.

You can replace the pauseus command with some other action. But remember if you take too much time it will slow down

your PWM frequency.

What kind of cable do I need to connect to the Dios carrier boards?

You will need a 9 pin male to female straight through cable. This will connect from the serial port on your PC to the Dios carrier board or EZRS232 driver.

Can the Dios clock frequency be changed?

The Dios chips are setup for 10Mhz resonator and are internally set to 40Mhz via a PLL.

This can not be changed.

I cant seem to get the low power when I put the Dios Chip to sleep. Any hints?

To gain the lowest power down mode you must place all unused leads into output mode and set them to high state. In Sleep mode you can get it all the way down to 4ua. This is pretty good for a 40Mhz chip.

For maximum shutdown turn off the UART with the RCSTA=0 command.

How much RAM is available on the Dios?

The total Ram on the Dios is 1536 bytes.

The Dios does not roll the memory into one flat memory model. For performance and operation it is divided up into the following segments

local memory

Used for local integer and floating point variable. With the memory model this memory is stacked based so as each function exits the memory is reclaimed. By using functions you get much more efficient use of memory. 256 bytes

Global memory

Used for integers and floating point variables. Accessible by all functions. This is memory similar to the Atom or Basic Stamp. 256 bytes

String memory

Used for strings. However if you don't use strings you can access this memory with the built in memread and memwrite commands 256 bytes

UART memory

Used by the built-in interrupt driven UART as a buffer. Again this memory can be used by your program for storage if the UART is not used. 256 bytes

Stack memory

Used by the Dios Engine to keep track of gosub and function call returns. This memory can not be used unless you don't wish to use other functions or gosubs. IE very similar to the flat model that the other high level language use. If you restrict yourself to that you can use most of this memory 256 bytes

Internal Engine memory

This is used by the internal engine. A great deal of this memory is available as single byte variables that you can use in your routines or in inline assembly routines. The registers BYTE0-BYTE25 and ASMDAT0-ASMDAT13 as well as others are available. 256 bytes

Can I add more memory to the Dios?

No you can not.

What is the maximum voltage the AtoD ports can take?

The Maximum voltage is Vdd. The voltage cannot rise above Vdd. If you need to measure a voltage higher then you will have to use resistors. See the Auto ranging Voltmeter Application note.

For measuring AC voltage you will have to use a Diode. You can not measure voltages that drop below Vss (negative).

If you have a complicated voltage requirement its is best to condition the signal with a op amp circuit.

What is the conversion time of the AtoD on the Dios?

The actual conversion time is based on the input capacitance and the voltage. Expect about 5000 times a second.

How do I use global constants in my inline assembly code?

Try this:

Make sure the gconst is defined inside Dios Code not assembly

'gconst with inline assembly example

```
func main()
  output 0
  gconst tmp1 7

startasm
loop:
  btg PORTB,GC_tmp1
  goto loop
endasm
```

```
endfunc
```

gconst is normaly used when you want both the Dios High Level language and assmby to have access to the constant.

If you only want the assembly code to have access to the constant use #define.

What circuit does the POT and Rctime expect?

Place a capacitor between port and Vss.

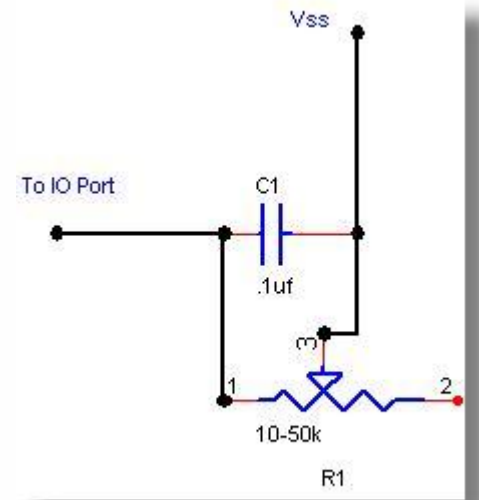
(different values for diff ranges)

Place a Pot between port and Vss.

Start with a .1 cap and 10-50k pot

However the rccount can be used to measure pulse width so depending on application you wont need a circuit.

There are code examples for both in the command_examples directory



If I call a subroutine with the gosub command can I exit a function while in that routine?

No this program will create a memory leak.

```
func main()
loop:
  tst
  goto loop

endfunc

func tst()

  gosub tmpy

tmpy:
  print "Abort"
  exit 1000

endfunc
```

The Stack space gets corrupted and the program will become unstable. You must pop the gosub location off the stack in order to exit the from the called gosub.

The following program shows you how to do this.

```
func main()
loop:
  tst
  goto loop
```

```
endfunc

func tst()

  gosub tmpy

tmpy:
  print "Abort"
  'Need to pop gosub stack before exit
  STACK1IDX = STACK1IDX - 2
  exit 1000

endfunc
```

Are there any increment and decrement short cuts?

Use the inc and dec commands.

Can I create my own default template for use when I create a new Dios file?

There is a file called template.txt located in the Dios directory.

This file is read each time you create a new file. It is used to give you a head start on each new program.

Feel free to modify this template. Just keep in mind that it will be overwritten when you upgrade the software so make a backup of any templates you plan on using.

Can the IO ports be pulled up or down internally?

Use the pullupon command. This will place an internal pullup resister on ports 0-7.

You can pull the ports low with buttons or other devices.

Is there a way to clear the serial UART?

When writing code that will use the hardware UART, make sure you use the hsersetup clear command to clear the buffer.

```
hsersetup clear
```

Can I create my own help files for my libraries?

Not unless you have a help compiler?

However....

IF you create your own library just save it into the lib directory with the .lib extension.

Create a text file with you instructions and give it a .txt extension. Just like the Dios Help Files your text file will pop up the library is double clicked in the library manager.

Bottom line... The library manager looks for a .hlp file first. If it can't find one it looks for a .txt file.

Is there a fast way to load and edit a include file?

You can quickly edit a include file by double clicking the include statement in you program file?

Can I drop files onto the Dios editor form?

You can drag a file from your desktop or other location and drop it onto a Dios Edit Form?

Its the same thing as loading a file with a dialog box. If the current file has not been saved you will be prompted to do so.

Is there a line continuation character available on the Dios Compiler?

You can use the _ to extend the line. This is important for creating readable code when using commands like lookup or lookdown.

lookup and lookdown support unlimited items. You don't want to do this in one line. Use the _ to break the line up into logical chunks.

Can I run more then one instance of the Dios Compiler?

With the creation of the CPslave library. It is very handy to be able to program two chips at the same time. You will have to have two available serial ports though.

The problem comes in when you try to communicate. As you can not have two chips connected to the same com port.

There are two ways to over come this.

1.
Start the Dios Editor. Then start another copy of the Dios Editor. On the second copy change the comport but don't save the settings. This will allow you to communicate with two different chips.

2.
Copy the Dios Editor short cut. Modify the properties of the second shortcut. In the target field add the parm loadcom x

X is the number of the com port to force at load time.

Is there pre/post increment/decrement math in dios language like in c?

No.

However there are inc and dec commands.

IE:
inc varb

This is faster and more efficient than doing the `varb = varb + 1`

Is there a way to see how many bytes there are waiting in the hserin buffer?

There are two registers:
RXHEAD and RXTAIL

Every time a byte comes in RXHEAD is bumped.

Every time you pull a byte off the buffer RXTAIL is bumped. If they are the same no bytes are in the buffer.

Keep in mind that they wrap at 256 so some if/then commands are needed to detect this.

You should be able to create a small function that returns the number of bytes in the buffer. Let us know what you come up with.

How do I determine how much variable space is available on the fly?

Local Variables

First let's look at how local memory is used on the Dios. It is used and freed in real time as the processor is being used. Because of this the compiler can not detect variable overflows. You can however use a couple of built in registers to calculate and test for local memory overflow.

The STACK2IDX and the CURVARBIDX registers contain the information the engine uses to calculate variables in real time.

STACK2IDX is an offset variable that indicates where a given function is supposed to begin local stack usage. IE the main function will always have a STACK2IDX of 0.

CURVARBIDX is used by the current function to keep track of how many variable units it is using. IE 1 for integers and 2 for float. The actual memory usage is $CURVARBIDX * 2$.

So using these two registers we can make the following test:
if $CURVARBIDX * 2 + STACK2IDX > 255$ then print "Overflow"

The following is a simple program to demonstrate this:

```
'simple real time memory test for local variables
func main()
  dim z(126) as integer
  if CURVARBIDX * 2 + STACK2IDX > 255 then print "Overflow main"
  mysub
endfunc

func mysub()
  dim a,b
  if CURVARBIDX * 2 + STACK2IDX > 255 then print "Overflow mysub"
endfunc
```

Global Variables

Global variables are a bit more strait forward. They are calculated during the compile. Again each integer variable uses 1 unit and float uses two units and each unit is 2 bytes. Global variables use there own data space and have 256 bytes available. Release 2.0.4 of the compiler will now give an error if you define too much global space.

If you use libraries like the LCDlibrary some global variable will be defined. All global variables will be listed on the compile form.

String Variables

String Variables use a there own data space and have 256 bytes available. A string uses the amount of memory allocated with the global command. Keep in mind that the last byte is used as a termination character.

Another note: If you are not using string variables you can read and write to this space using the memread and memwrite commands.

Can the resolution of the pulsein command be changed?

The register value PULSEINSCALE will allow you to change the resolution of the pulsein command.

The default value is 8 which sets the resolution to .1us per unit. Use the following table to set the resolution.

0	- .2us per unit
1	- .4us per unit
2	- .8us per unit
3	- 1.6us per unit
4	- 3.2us per unit
5	- 6.4us per unit
6	- 12.8us per unit

What determines the state of the pulseout command?

The pulseout just toggles what ever state its already in.

If it is high then it will go low for xxxus then back high.

If it is low the then it will go high for xxxus then back low.

The time between the calls will determine the duty cycle.

```
loop:
  pulseout 0,100
  pauseus 100
  goto loop
```

In theory this should give you a 50% duty cycle but the latency in the goto command must be added as well.

If a hardware IRQ is generated while the my Dios program is using the pause command what happens.

The Dios has two types of IRQ handling, software and hardware.

In general all IRQ's are hardware and will interrupt any thing, pulsein, pause or even debug or print.

This is why you can get garbage when printing and heavy IRQ's are being triggered.

The onirq handlers when are very simple it just sets a couple of flags. This is where the software IRQs come in. At the point just before each command call a test is made to see if a IRQ fired. If you have set up a software IRQ handler with the onirq command a function call will be made to your handler.

Using the asmirq command you can write a hardware handler. These are called as soon as the IRQ is detected.

All IRQ's will cause some disruption to timeing orented commands like pulsein and pause and serout commands.

Can I use the onirq command with timer1,timer2, and timer3?

Timer1,Timer2 and Timer3 all work pretty much the same as timer0. However Timer 0 has the most robustness.

Timer 1 is a 8 or 16 bit timer the prescaler does not have the resolution of timer0. Timer 1 can also be driven by an external 32Khz crystal to create a clock id desired.

Timer 2 is a very simple 8 bit timer with a limited prescaler.

Timer 3 is pretty much identical to Timer1.

They are all set up about the same as timer0 but the register names and bits are different. The Timer Primer application note shows how to do an onirq with a timer.

What does FOSC/4 mean?

FOSC/4 Is Main Oscillator /4 in this case 40Mhz/10 so its 10Mhz.

Does the Dios Language support parentheses in math expressions?

In order to keep the math and condition testing as fast as possible I had to make some sacrifices.

One of those sacrifices is to not support parentheses. So how does it work?

Math expressions are handled as they are encountered. Internally there is an internal accumulator used in the math and expression routines.

Lets look at the following.

$X = 10 + 90 * 5$

step1 10 is placed in accumulator, accumulator = 10

step2 90 is added to accumulator, accumulator = 100

step3 accumulator is multiplied by 5, accumulator = 500

step4 result is place in variable X, X = 500

It works like the way you use a simple calculator. If you need to do some real fancy math just break it down into multiple statements and multiple variables. This will make your code more readable in the long run.

What floating point format does the Dios use?

The Dios Language uses IEEE 754 32bit format.

It is slightly modified, the Sign bit is moved to the 23rd bit instead of the 31st bit.

It handles the exception conditioning as well.

Does the Dios support recursion? If so how deep?

Yes the Dios functions support recursion. The depth depends upon how much stack space you are using when you make the call.

There is 256 bytes of stack space available. Each function call takes 7 bytes of stack space. Also note that if you are using local variables they come off of the variable stack which also has 256 bytes. Each local integer variable uses 2 bytes and each local float variable takes 4 bytes.

Is there a way of converting a string valu to an integer value similar to the val function in VB?

In the DiosString library is a function called STRval. This will convert to an integer.

Also in the GPS application note there is a function called GPSfloatval that will convert a string to floating point.

How do I shift bits right or left with the Dios language?

Just multiply or divide the variable by 2,4,8,16,32,64 and so on.

myvarb = myvarb / 2 Shift right 1 bit

myvarb = myvarb * 2 Shift left 1 bit

Does the Dios Language Support signed numbers in its variables?

Integer variables are 16 bit unsigned variables. If you need to do work with signed math use the floating point variables.

Dios floating point math is very fast.