

Programming Arduino

Can I use int main() with Arduino

You can provide a custom int main() function to be used in place of the default which is provided by the Arduino API, however providing your own in the sketch will override the built in version.

There are a few things to take note before using it as there are a few things you may need to do depending on the requirements of your sketch. I have created a few points below which highlight a few specifics to consider.

- **Do not return from main()**

As there is no operating system running on a standard Arduino^[1] there is no where to return to. If you do return, abort() is called, then exit(). On the AVR-8 MCU (Uno, Mega) this results in global interrupts being disabled and execution enters an infinite loop.

To overcome the eventual return, you can utilize an infinite loop of your own, which is not unlike what the Arduino core does itself.

- **Call the init() function**

If you use parts of the API such as PWM control and the millis() timer, you will need to call the core function init() which will setup the microcontroller for use with the Arduino core API features. It must be called before the first use of the features it initialises.

I will update this list with the exact features requiring init() to be called.

- PWM
- millis()
- micros()
- delay() <- uses micros

- **Start USB interface.**

A selection of Arduino boards that use the ATmega 32u4 processor, like the Leonardo, require their USB controller to be running. It is also required for serial communication.

The USB features can be controlled by a define which is only present on applicable boards.

```
#if defined( USBCON )
    USBDevice.attach();
#endif
```

A typical main() function is included below, it is also a complete replica of the Arduino's version, the only minor difference is the use of a while loop over a for loop.

Programming Arduino

```
int main( void ){

    //Initialise Arduino functionality.
    init();

    //Attach USB for applicable processors.
    #ifdef USBCON
        USBDevice.attach();
    #endif

    setup(); //Call Arduino standard setup function

    while( true ){

loop(); //Call Arduino standard loop function.

//Process the serial libraries event queue.
if( serialEventRun ) serialEventRun();
    }
    //Execution should never reach this point.
    return 0x00;
}
```

Tag notes.

1. You can install an operating system like an RTOS on your Arduino, this is covered in a future answer.

Unique solution ID: #1001

Author: Christopher Andrews

Last update: 2013-12-31 09:49