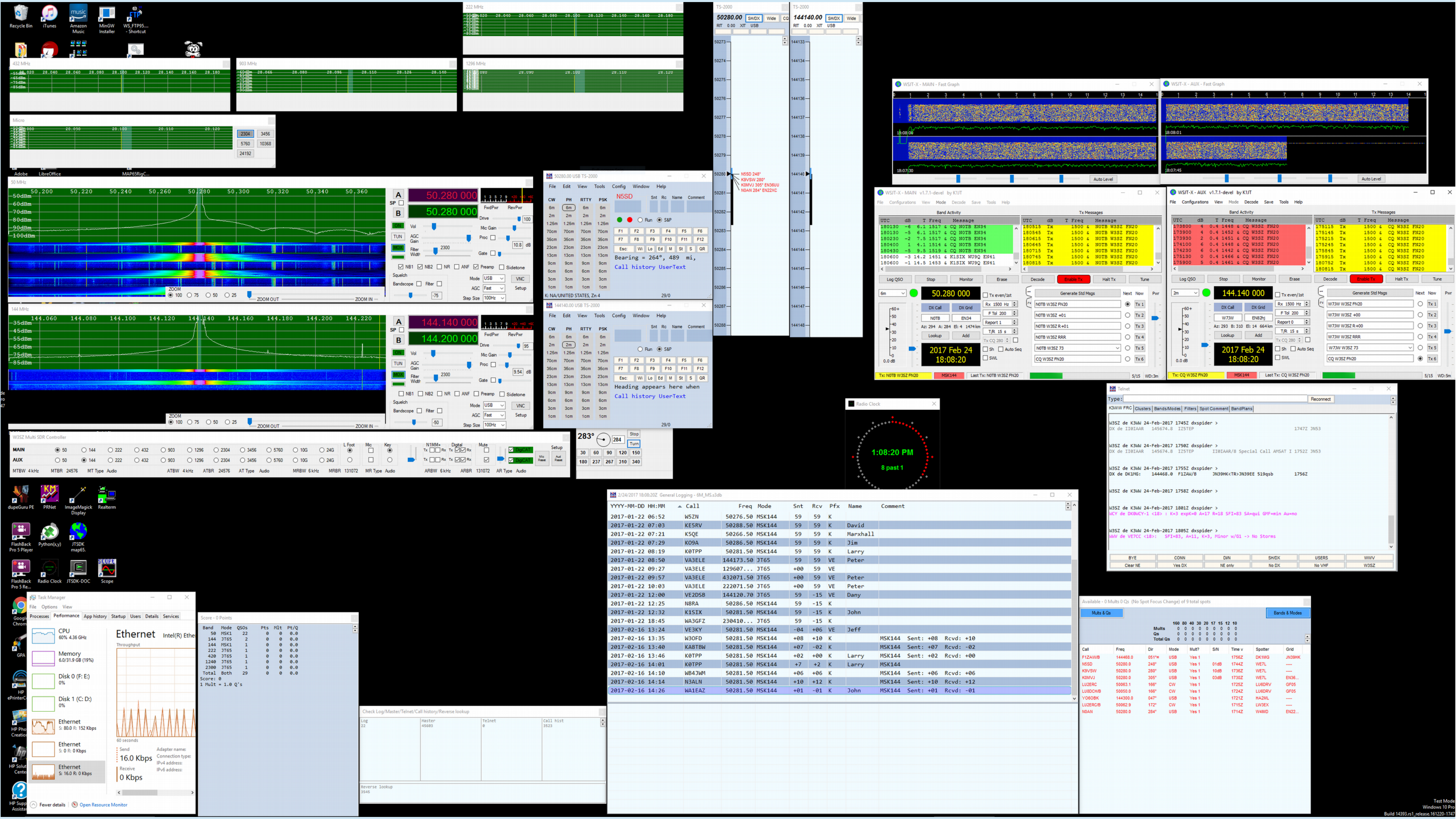
Station Automation --W3SZ



2/24/2017 18:08:20Z General Logging - 6M_MS.s3db

```
YYYY-MM-DD HH:MM   Call        Freq  Mode    Snt  Rcv  Pfx  Name      Comment
2017-01-22 06:52   W5ZN     50276.50 MSK144  59   59   K
2017-01-22 07:03   KE5RV    50288.50 MSK144  59   59   K    David
2017-01-22 07:21   K5QE     50266.50 MSK144  59   59   K    Marxhall
2017-01-22 07:29   KO9A     50286.50 MSK144  59   59   K    Jim
2017-01-22 08:19   K0TPP    50281.50 MSK144  59   59   K    Larry
2017-01-22 08:50   VA3ELE  144173.50 JT65    59   59   VE   Peter
2017-01-22 09:27   VA3ELE  129607... JT65    +00  59   VE
2017-01-22 09:57   VA3ELE  432071.50 JT65    +00  59   VE   Peter
2017-01-22 10:03   VA3ELE  222071.50 JT65    +00  59   VE   Peter
2017-01-22 12:00   VE2DSB  144120.70 JT65    59   -15  VE   Dany
2017-01-22 12:25   N8RA     50286.50 MSK144  59   -15  K
2017-01-22 12:32   K1SIX    50281.50 MSK144  59   -15  K    John
2017-01-22 18:45   WA3GFZ  230410... JT65    59   -15  K
2017-02-16 13:24   VE3KY    50281.50 MSK144  -04  +06  VE   Jeff
2017-02-16 13:35   W3OFD    50281.50 MSK144  +08  +10  K              MSK144  Sent: +08  Rcvd: +10
2017-02-16 13:40   KA8TBW   50281.50 MSK144  +07  -02  K              MSK144  Sent: +07  Rcvd: -02
2017-02-16 13:46   K0TPP    50281.50 MSK144  +02  +00  K    Larry     MSK144  Sent: +02  Rcvd: +00
2017-02-16 14:01   K0TPP    50281.50 MSK144  +7   +2   K    Larry     MSK144
2017-02-16 14:10   WB4JWM   50281.50 MSK144  +06  +06  K              MSK144  Sent: +06  Rcvd: +06
2017-02-16 14:14   N3ALN    50281.50 MSK144  +10  +12  K              MSK144  Sent: +10  Rcvd: +12
2017-02-16 14:26   WA1EAZ   50281.50 MSK144  +01  -01  K    John      MSK144  Sent: +01  Rcvd: -01
```

Telnet

```
W3SZ de K3WW 24-Feb-2017 1745Z dxspider >
DX de II0IAAR   145674.8  IZ5TEP                            1747Z JN53

W3SZ de K3WW 24-Feb-2017 1750Z dxspider >
DX de II0IAAR   145674.8  IZ5TEP       II0IAAR/8 Special Call AMSAT I 1752Z JN53

W3SZ de K3WW 24-Feb-2017 1755Z dxspider >
DX de DK1MG:    144468.0  F1ZAW/B      JN39HK<TR>JN39EE 519qsb       1756Z

W3SZ de K3WW 24-Feb-2017 1758Z dxspider >

W3SZ de K3WW 24-Feb-2017 1801Z dxspider >
WCY de DK0WCY-1 <18> : K=3 expK=0 A=17 R=18 SFI=83 SA=qui GMF=min Au=no

W3SZ de K3WW 24-Feb-2017 1805Z dxspider >
WWV de VE7CC <18>:   SFI=83, A=11, K=3, Minor w/G1 -> No Storms
```

Now Back to Previously Scheduled Program

USB-Serial IF/Transverter Bandswitch

Arduino-VHFLog Example

- I started with Ed Finn WA3DRC's excellent code that was written to give TS2000 CAT control via USB-Serial data from VHFLog
- Removed portions of Ed's code that deal only with CAT control of TS2000
- Modified / Added to remaining portions of Ed's code so that band switching in VHFLog bandswitches hardware by activating a separate relay for each band 50 MHz through 76 GHz instead of sending CAT data to TS2000

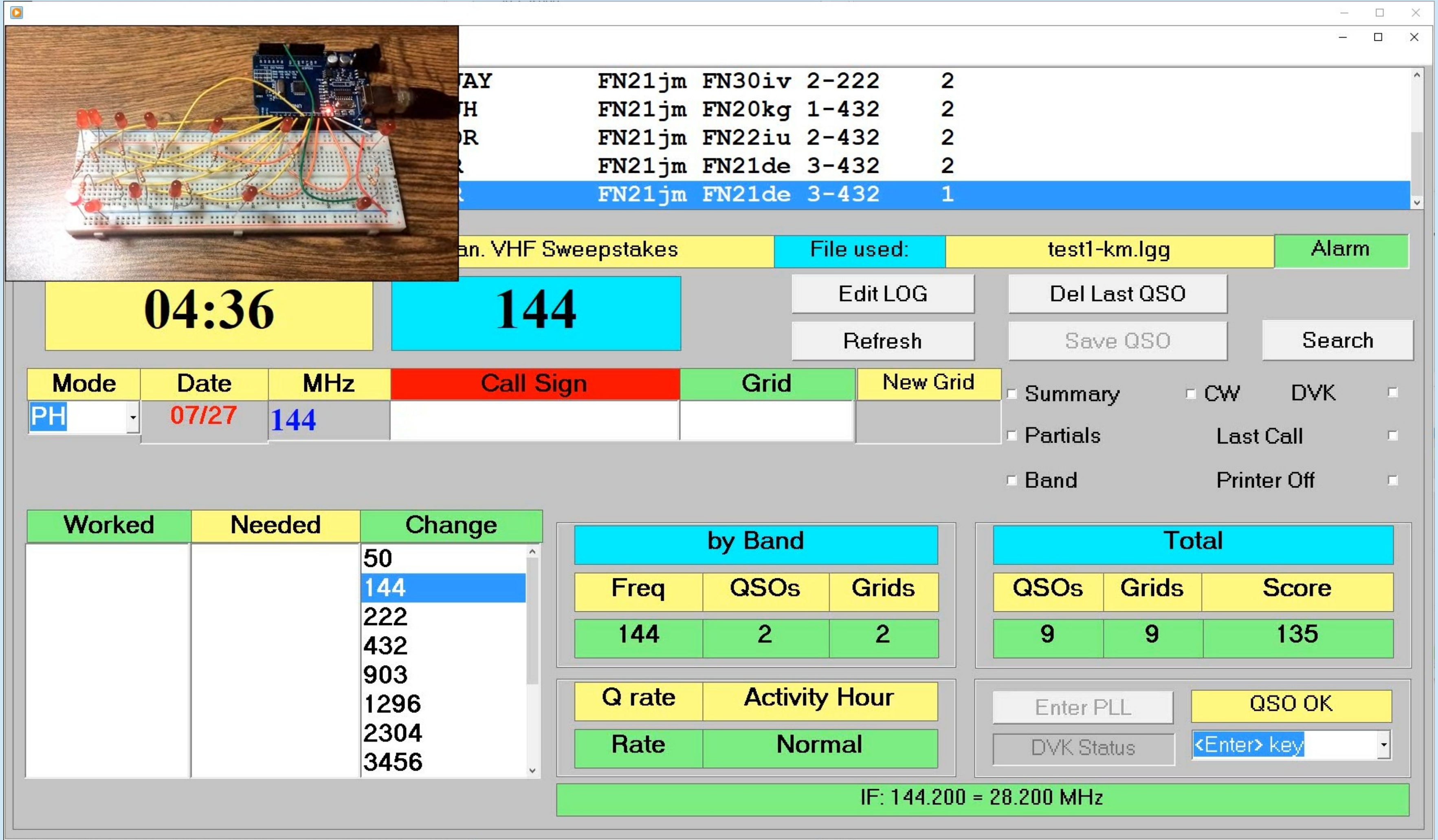
Arduino Example

USB-Serial Bandswitching from VHFLog

- Ed’s code is here:
  - http://w3sz.com/ts2000_vhflog_mega.ino
- Modifed code is here:
  - http://w3sz.com/vhflog_SainSmart_14CH_UNO.ino

IF/TransverterBand Switching - USB-Serial

- VHFLog will also work with Arduino
  - Either real serial port or USB-serial port



```
AY        FN21jm FN30iv 2-222    2
H         FN21jm FN20kg 1-432    2
R         FN21jm FN22iu 2-432    2
          FN21jm FN21de 3-432    2
          FN21jm FN21de 3-432    1
```

an. VHF Sweepstakes | File used: | test1-km.lgg | Alarm

04:36 | 144

Edit LOG | Del Last QSO | Refresh | Save QSO | Search

Mode	Date	MHz	Call Sign	Grid	New Grid
PH	07/27	144			

□ Summary □ CW DVK □
□ Partials Last Call □
□ Band Printer Off □

Worked	Needed	Change
		50 144 222 432 903 1296 2304 3456

by Band		
Freq	QSOs	Grids
144	2	2

Total		
QSOs	Grids	Score
9	9	135

Q rate	Activity Hour
Rate	Normal

Enter PLL | QSO OK
DVK Status | <Enter> key

IF: 144.200 = 28.200 MHz

IF/TransverterBand Switching - USB-Serial

E:\StationAutomation\PackRatsMiniTalk\1_VHFLogPIP-2.wmv

Arduino Example

- VHFLog sends
  - “50” for 50 MHz band
  - “14” for 144 MHz band
  - “22” for 222 MHz band
  - “”43” for 432 MHz band
  - “90” for 902 MHz band
  - “12” for 1296 MHz band
  - “23” for 2304 MHz band
  - “34” for 3456 MHz band
  - “57” for 5760 MHz band
  - “10” for 10368 MHz band
  - “24” for 24192 MHz band
  - “47” for 47000 MHz band
  - “76” for 76000 MHz band

1) Capture two digit band info from serial port
2) Parse it
3) Use it to switch bands by setting to “High” the output pin assigned to the selected band, while making sure all other output pins are set to “Low”

This will be a **small** program with **13 output pins** needed (one for each band)

Getting Started With Arduino

- **Choose which Arduino to use** based on:
  - Number of GPIO pins needed **(HERE 13)**
  - Amount of memory needed (SRAM and flash)
    - **HERE “small” program**

UNO

Getting Started With Arduino

- **Download Arduino IDE**
  - https://www.arduino.cc/en/Main/Software
- Or use the online Arduino Web Editor
  - https://www.arduino.cc/en/Main/Software
  - Need to create an Arduino Account
  - Read instructions at:
    - https://create.arduino.cc/projecthub/Arduino_Genuino/getting-started-with-the-arduino-web-editor-4b3e4a
  - Log on and get started, following the instructions

Arduino

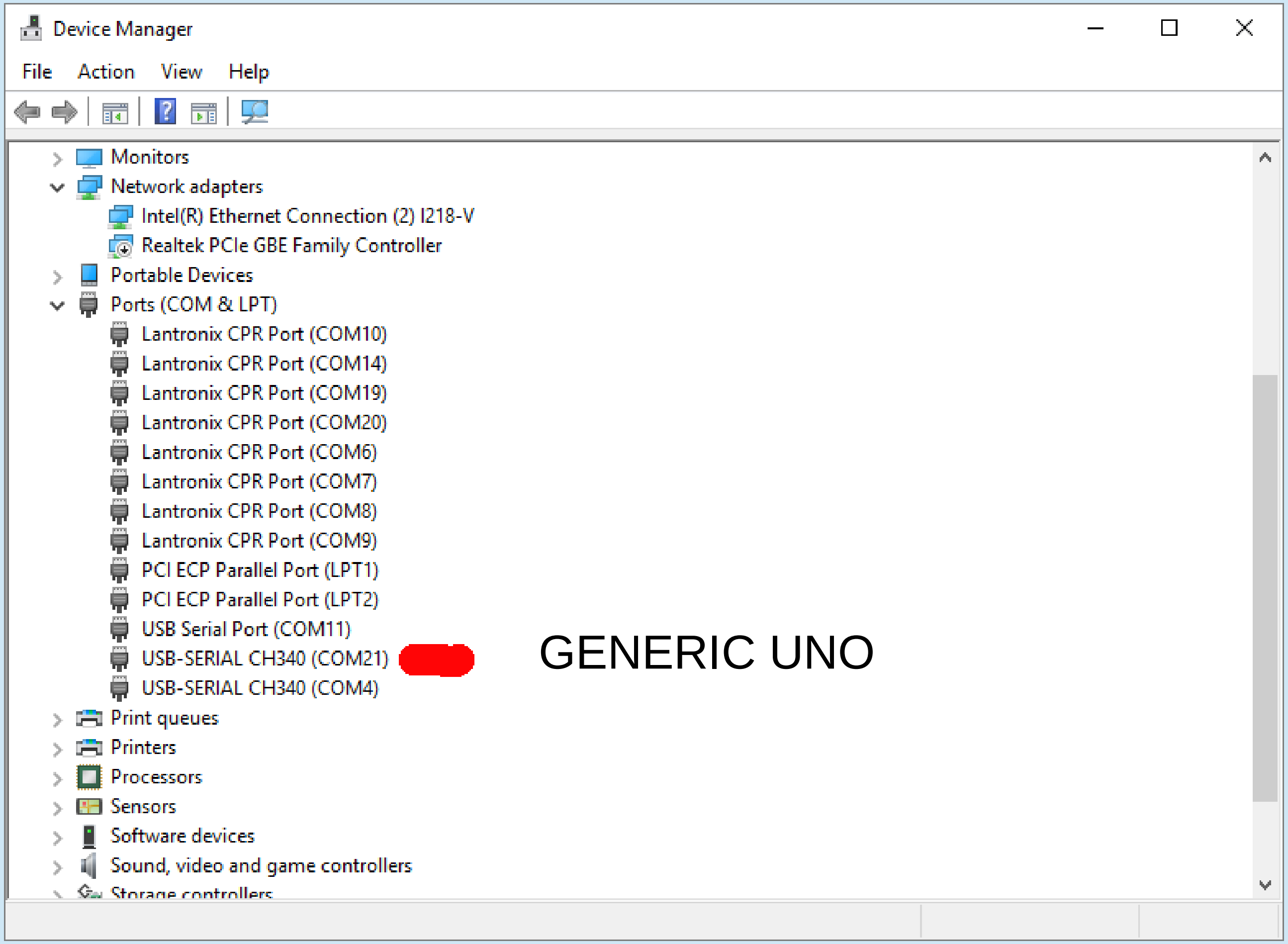
- Arduino language is based on C/C++
- Links against the AVR Libc and allows use of all Libc functions
  - http://www.nongnu.org/avr-libc/
- Language reference is here:
  - https://www.arduino.cc/en/Reference/HomePage
- There are Arduino-specific Libraries for extending language
  - Ethernet, WiFi, I2C, Stepper, SPI, Servo, SD, LiquidCrystal, EEPROM, Debounce, FFT, etc.
  - https://www.arduino.cc/en/Reference/Libraries
- There is an excellent Arduino tutorial here:
  - https://www.tutorialspoint.com/arduino/index.htm

Arduino

- There is an active **forum** at:
  - https://forum.arduino.cc/
- The **Auduino Playground** has many open-source **sketches** (programs) that you can copy, modify, use:
  - https://playground.arduino.cc/
- There is an **Arduino StackExchange!**
  - https://arduino.stackexchange.com/
- There is more helpful stuff on the Arduino than you would be able to read if you start now and don’t stop until the day you die!
- No matter what you want to do, it is likely that someone else has done something like it. So don’t reinvent the wheel! Start with their code and modify it as necessary. DO THIS FOR EVERY BUILDING BLOCK IN YOUR CODE AND THINGS WILL MOVE VERY QUICKLY FOR YOU (and you will learn more than you ever would by reading textbooks)!!
- **Google is your (BEST) Friend!!**

Arduino
Getting Ready

- Download the IDE and Install it. That should also install the driver.

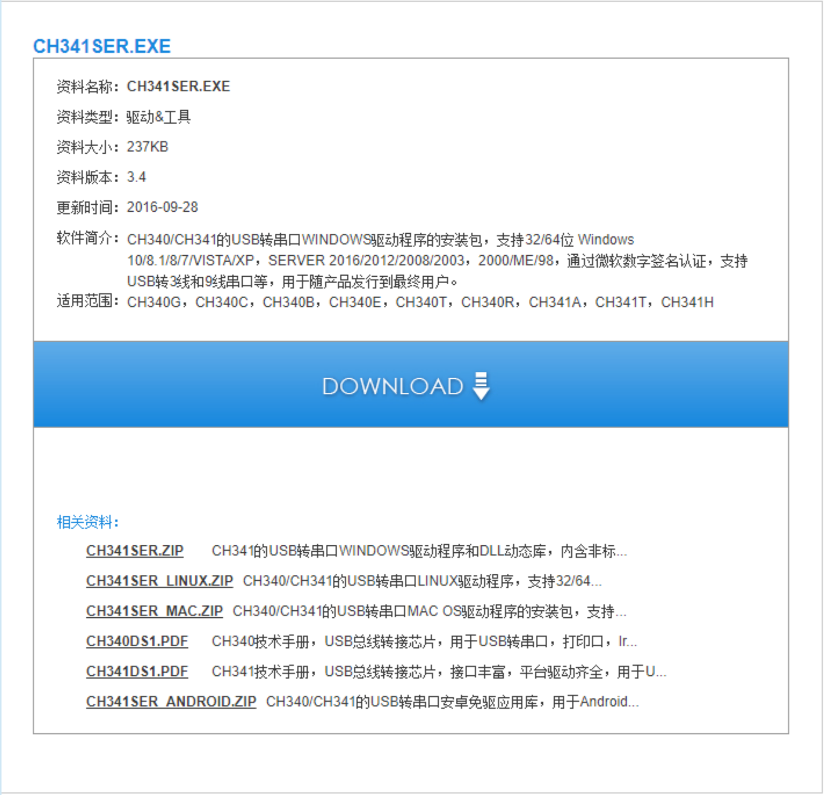


- Monitors
- Network adapters
  - Intel(R) Ethernet Connection (2) I218-V
  - Realtek PCIe GBE Family Controller
- Portable Devices
- Ports (COM & LPT)
  - Lantronix CPR Port (COM10)
  - Lantronix CPR Port (COM14)
  - Lantronix CPR Port (COM19)
  - Lantronix CPR Port (COM20)
  - Lantronix CPR Port (COM6)
  - Lantronix CPR Port (COM7)
  - Lantronix CPR Port (COM8)
  - Lantronix CPR Port (COM9)
  - PCI ECP Parallel Port (LPT1)
  - PCI ECP Parallel Port (LPT2)
  - USB Serial Port (COM11)
  - USB-SERIAL CH340 (COM21)
  - USB-SERIAL CH340 (COM4)
- Print queues
- Printers
- Processors
- Sensors
- Software devices
- Sound, video and game controllers
- Storage controllers

GENERIC UNO

Arduino
Getting Ready

- Download the IDE and Install it. That should also install the driver. If it doesn’t, download and install drivers from http://www.wch.cn/download/CH341SER_EXE.html



CH341SER.EXE

资料名称：CH341SER.EXE

资料类型：驱动&工具

资料大小：237KB

资料版本：3.4

更新时间：2016-09-28

软件简介：CH340/CH341的USB转串口WINDOWS驱动程序的安装包，支持32/64位 Windows 10/8.1/8/7/VISTA/XP，SERVER 2016/2012/2008/2003，2000/ME/98，通过微软数字签名认证，支持USB转3线和9线串口等，用于随产品发行到最终用户。

适用范围：CH340G，CH340C，CH340B，CH340E，CH340T，CH340R，CH341A，CH341T，CH341H

相关资料：

CH341SER.ZIP CH341的USB转串口WINDOWS驱动程序和DLL动态库，内含非标...

CH341SER_LINUX.ZIP CH340/CH341的USB转串口LINUX驱动程序，支持32/64...

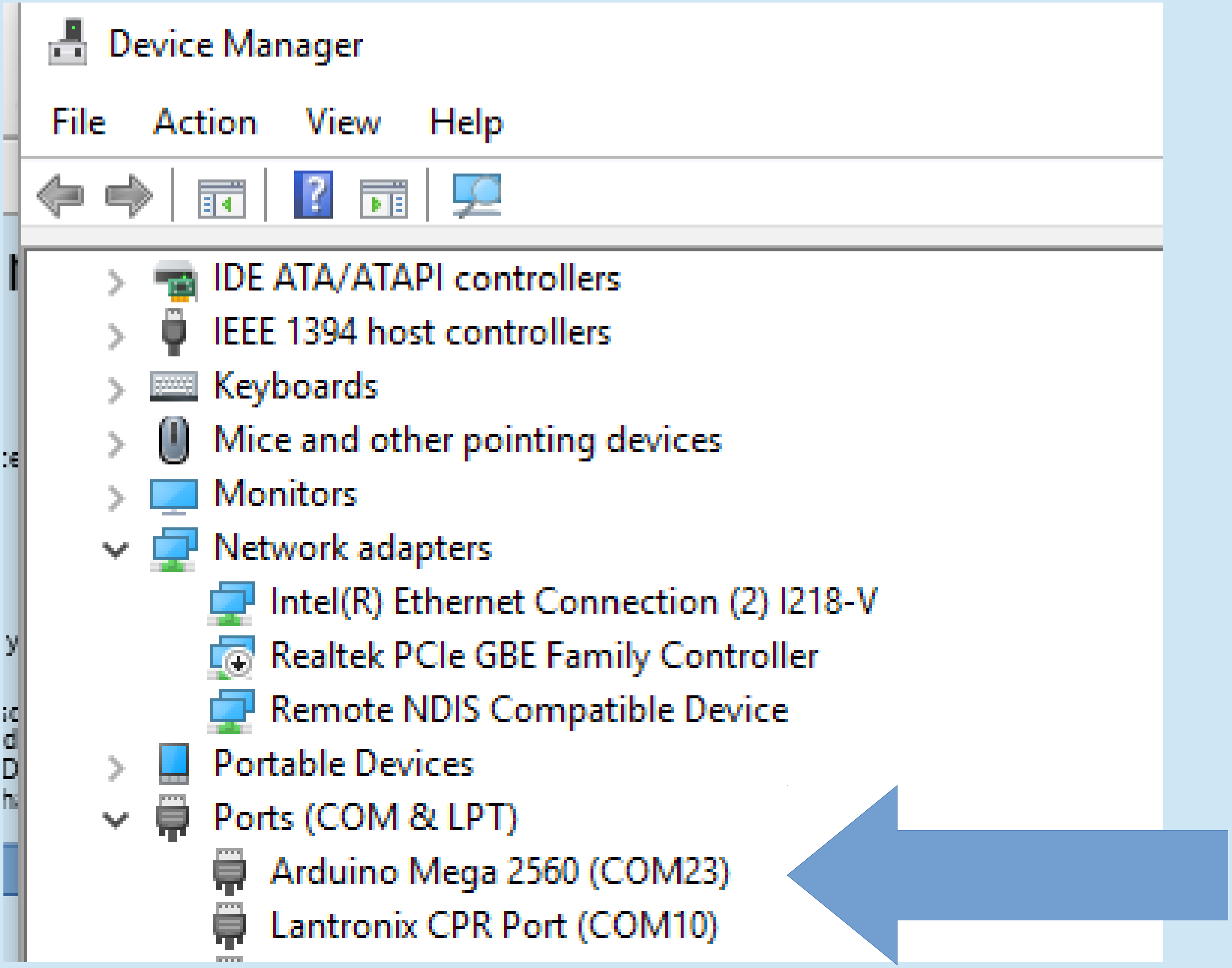
CH341SER_MAC.ZIP CH340/CH341的USB转串口MAC OS驱动程序的安装包，支持...

CH340DS1.PDF CH340技术手册，USB总线转接芯片，用于USB转串口，打印口，Ir...

CH341DS1.PDF CH341技术手册，USB总线转接芯片，接口丰富，平台驱动齐全，用于U...

CH341SER_ANDROID.ZIP CH340/CH341的USB转串口安卓免驱应用库，用于Android...

Arduino
Getting Ready

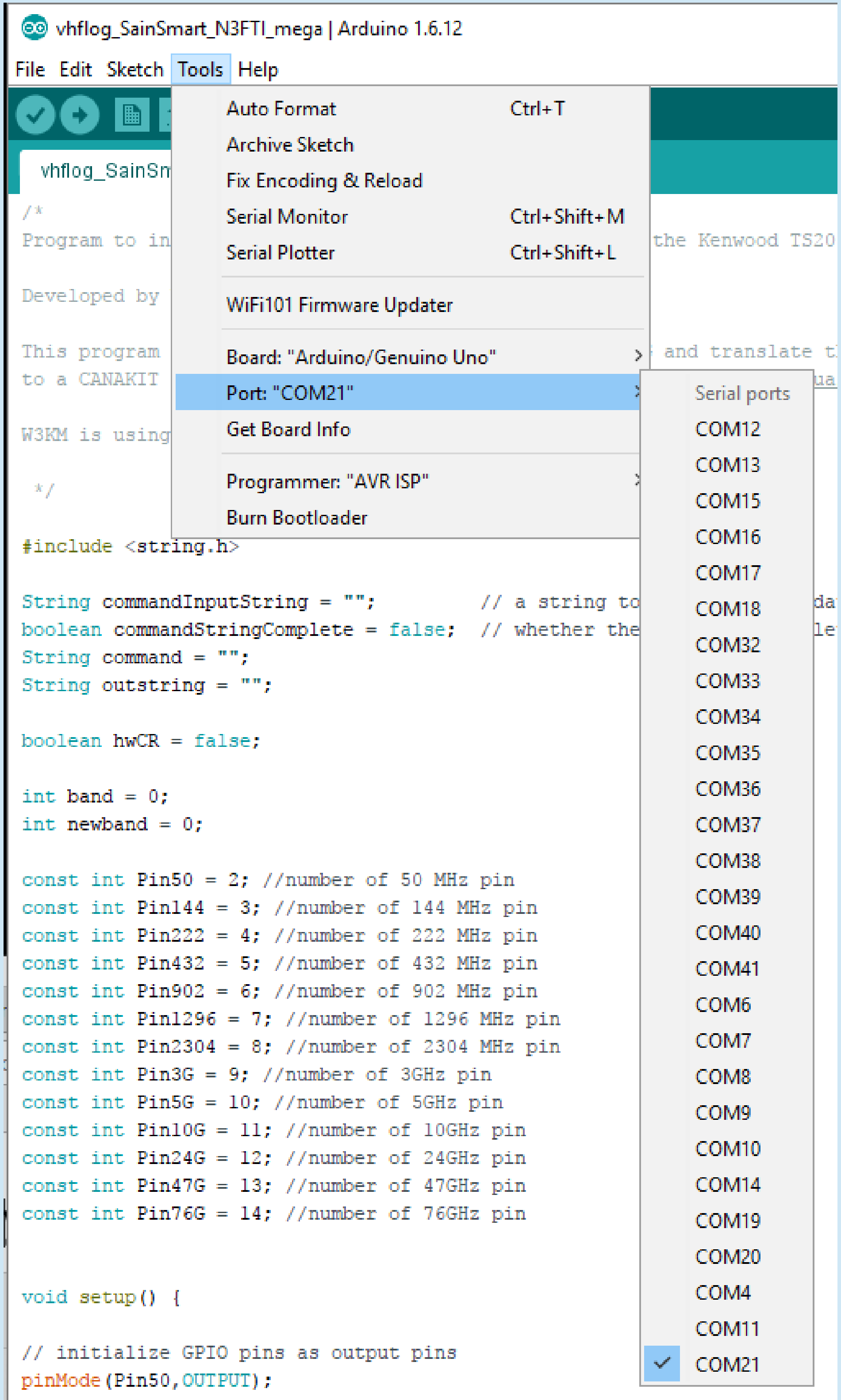


Device Manager

- IDE ATA/ATAPI controllers
- IEEE 1394 host controllers
- Keyboards
- Mice and other pointing devices
- Monitors
- Network adapters
  - Intel(R) Ethernet Connection (2) I218-V
  - Realtek PCIe GBE Family Controller
  - Remote NDIS Compatible Device
- Portable Devices
- Ports (COM & LPT)
  - Arduino Mega 2560 (COM23)
  - Lantronix CPR Port (COM10)

Arduino
Getting Ready

- Start the IDE
- Select COM port



Board: "Arduino/Genuino Uno"

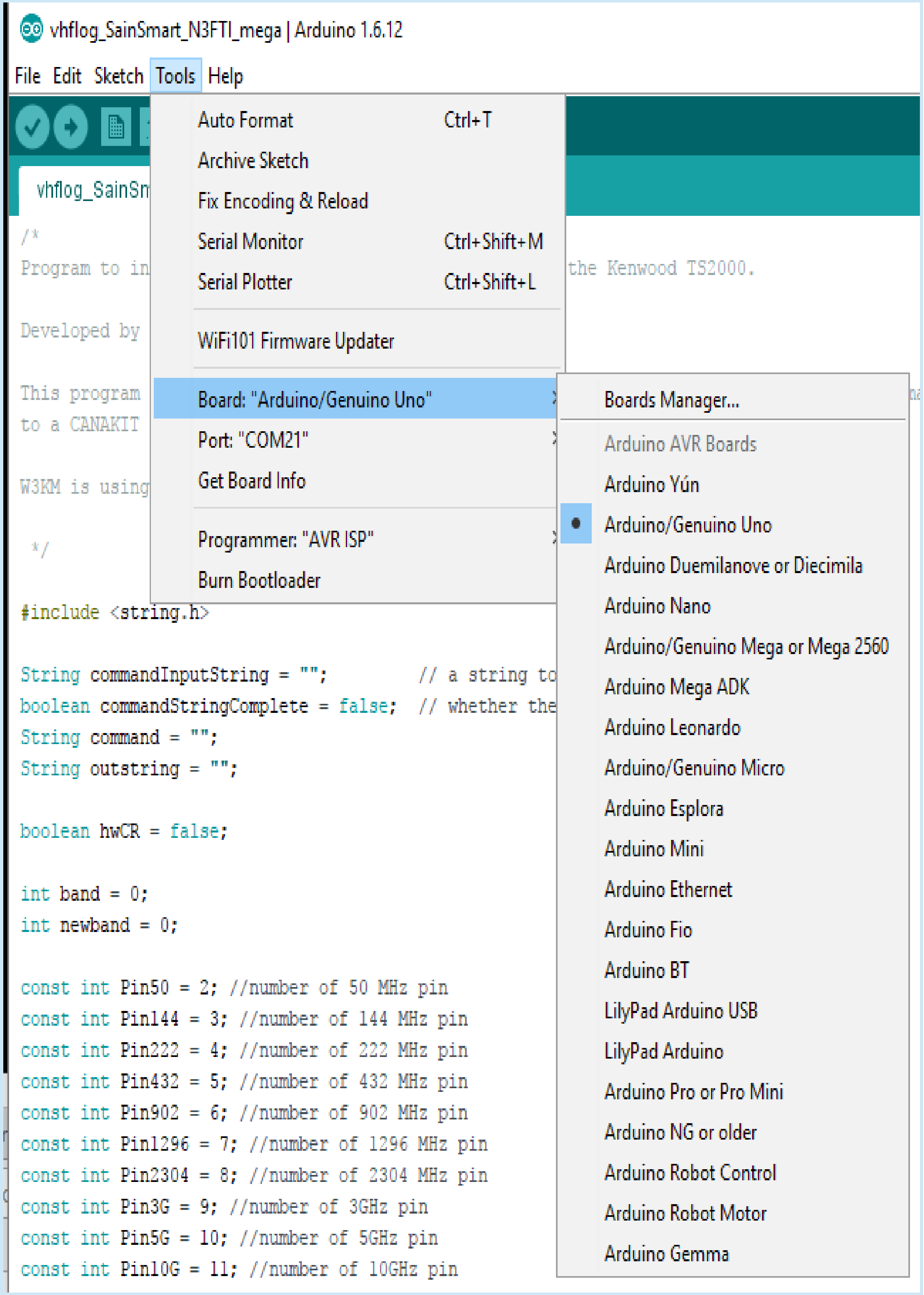
Port: "COM21"

Programmer: "AVR ISP"

```
/*
Program to in                                    the Kenwood TS20

Developed by

This program                                      and translate t
to a CANAKIT                                                     ua

W3KM is using

 */

#include <string.h>

String commandInputString = "";         // a string to                da
boolean commandStringComplete = false;  // whether the                le
String command = "";
String outstring = "";

boolean hwCR = false;

int band = 0;
int newband = 0;

const int Pin50 = 2; //number of 50 MHz pin
const int Pin144 = 3; //number of 144 MHz pin
const int Pin222 = 4; //number of 222 MHz pin
const int Pin432 = 5; //number of 432 MHz pin
const int Pin902 = 6; //number of 902 MHz pin
const int Pin1296 = 7; //number of 1296 MHz pin
const int Pin2304 = 8; //number of 2304 MHz pin
const int Pin3G = 9; //number of 3GHz pin
const int Pin5G = 10; //number of 5GHz pin
const int Pin10G = 11; //number of 10GHz pin
const int Pin24G = 12; //number of 24GHz pin
const int Pin47G = 13; //number of 47GHz pin
const int Pin76G = 14; //number of 76GHz pin


void setup() {

// initialize GPIO pins as output pins
pinMode(Pin50,OUTPUT);
```

Arduino
Getting Ready

- Start the IDE
- Select COM port
- Select Arduino Type



Board: "Arduino/Genuino Uno"

Port: "COM21"

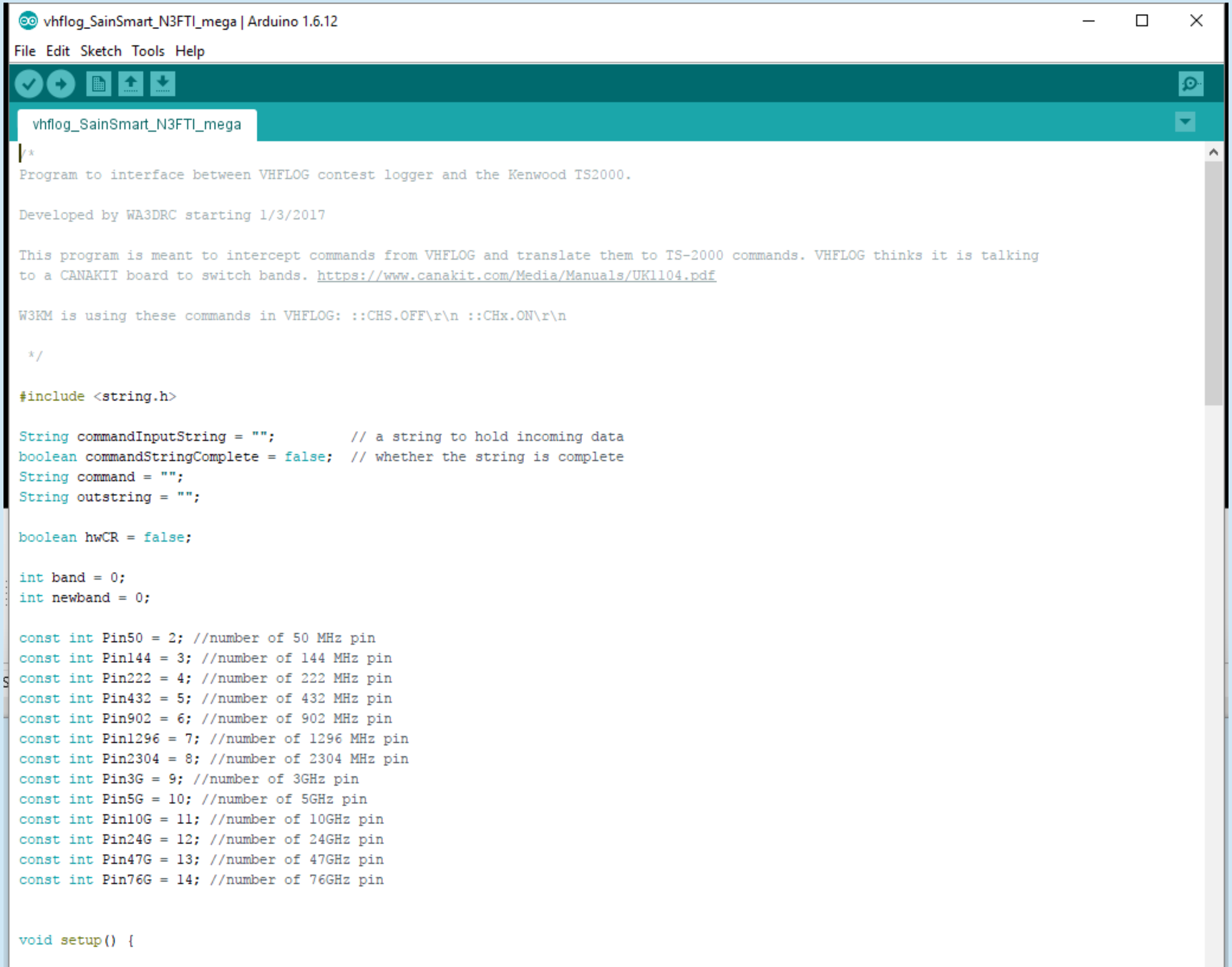
Programmer: "AVR ISP"

● Arduino/Genuino Uno

```
/*
Program to in                                    the Kenwood TS2000.

Developed by

This program
to a CANAKIT

W3KM is using

 */
#include <string.h>

String commandInputString = "";         // a string to
boolean commandStringComplete = false;  // whether the
String command = "";
String outstring = "";

boolean hwCR = false;

int band = 0;
int newband = 0;

const int Pin50 = 2; //number of 50 MHz pin
const int Pin144 = 3; //number of 144 MHz pin
const int Pin222 = 4; //number of 222 MHz pin
const int Pin432 = 5; //number of 432 MHz pin
const int Pin902 = 6; //number of 902 MHz pin
const int Pin1296 = 7; //number of 1296 MHz pin
const int Pin2304 = 8; //number of 2304 MHz pin
const int Pin3G = 9; //number of 3GHz pin
const int Pin5G = 10; //number of 5GHz pin
const int Pin10G = 11; //number of 10GHz pin
```

Arduino
Getting Ready

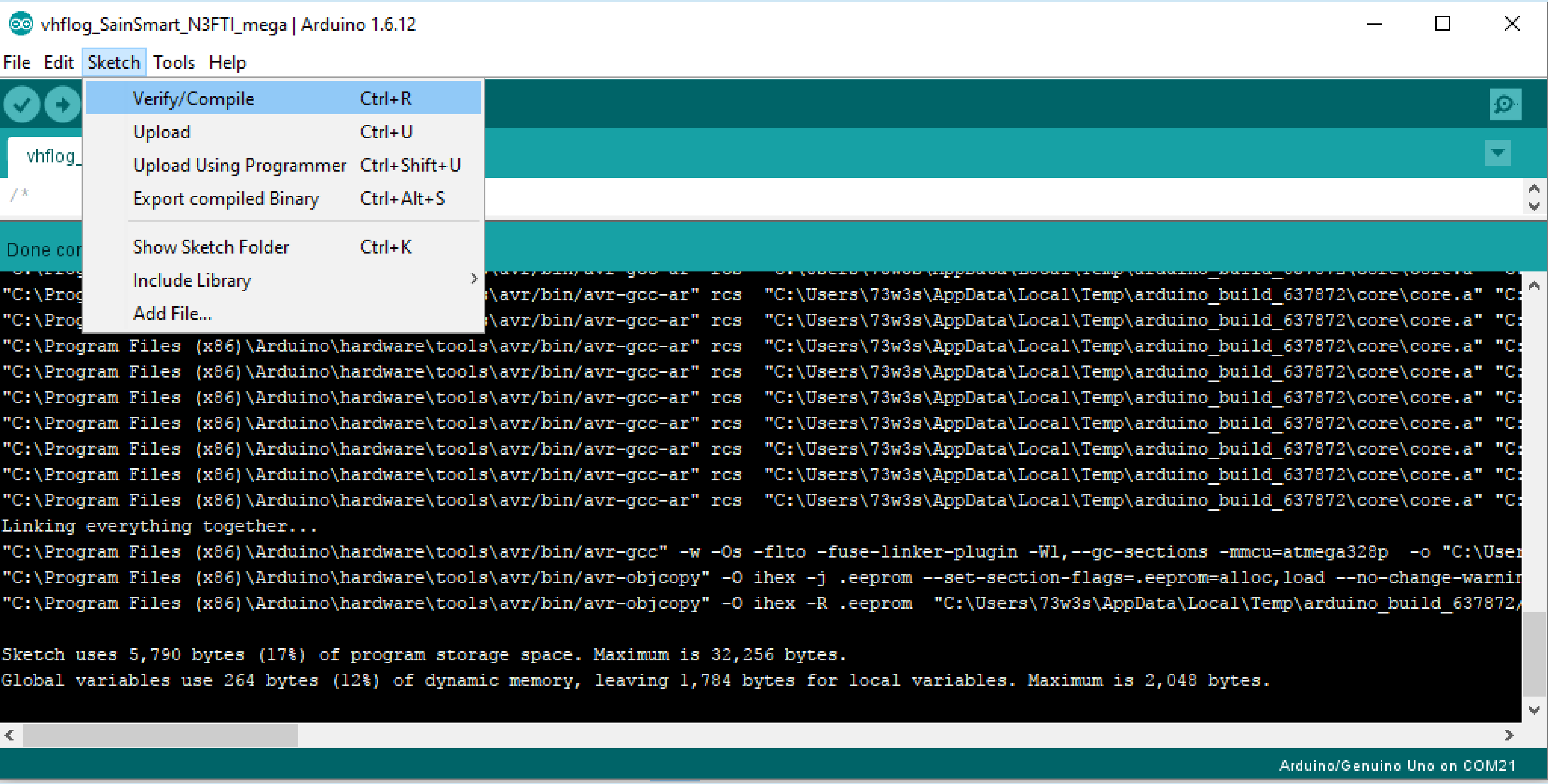
- Write the code (“Sketch”)


vhflog_SainSmart_N3FTI_mega | Arduino 1.6.12
File Edit Sketch Tools Help
vhflog_SainSmart_N3FTI_mega

```
/*
Program to interface between VHFLOG contest logger and the Kenwood TS2000.

Developed by WA3DRC starting 1/3/2017

This program is meant to intercept commands from VHFLOG and translate them to TS-2000 commands. VHFLOG thinks it is talking
to a CANAKIT board to switch bands. https://www.canakit.com/Media/Manuals/UK1104.pdf

W3KM is using these commands in VHFLOG: ::CHS.OFF\r\n ::CHx.ON\r\n

 */

#include <string.h>

String commandInputString = "";         // a string to hold incoming data
boolean commandStringComplete = false;  // whether the string is complete
String command = "";
String outstring = "";

boolean hwCR = false;

int band = 0;
int newband = 0;

const int Pin50 = 2; //number of 50 MHz pin
const int Pin144 = 3; //number of 144 MHz pin
const int Pin222 = 4; //number of 222 MHz pin
const int Pin432 = 5; //number of 432 MHz pin
const int Pin902 = 6; //number of 902 MHz pin
const int Pin1296 = 7; //number of 1296 MHz pin
const int Pin2304 = 8; //number of 2304 MHz pin
const int Pin3G = 9; //number of 3GHz pin
const int Pin5G = 10; //number of 5GHz pin
const int Pin10G = 11; //number of 10GHz pin
const int Pin24G = 12; //number of 24GHz pin
const int Pin47G = 13; //number of 47GHz pin
const int Pin76G = 14; //number of 76GHz pin


void setup() {
```

Arduino
Getting Ready

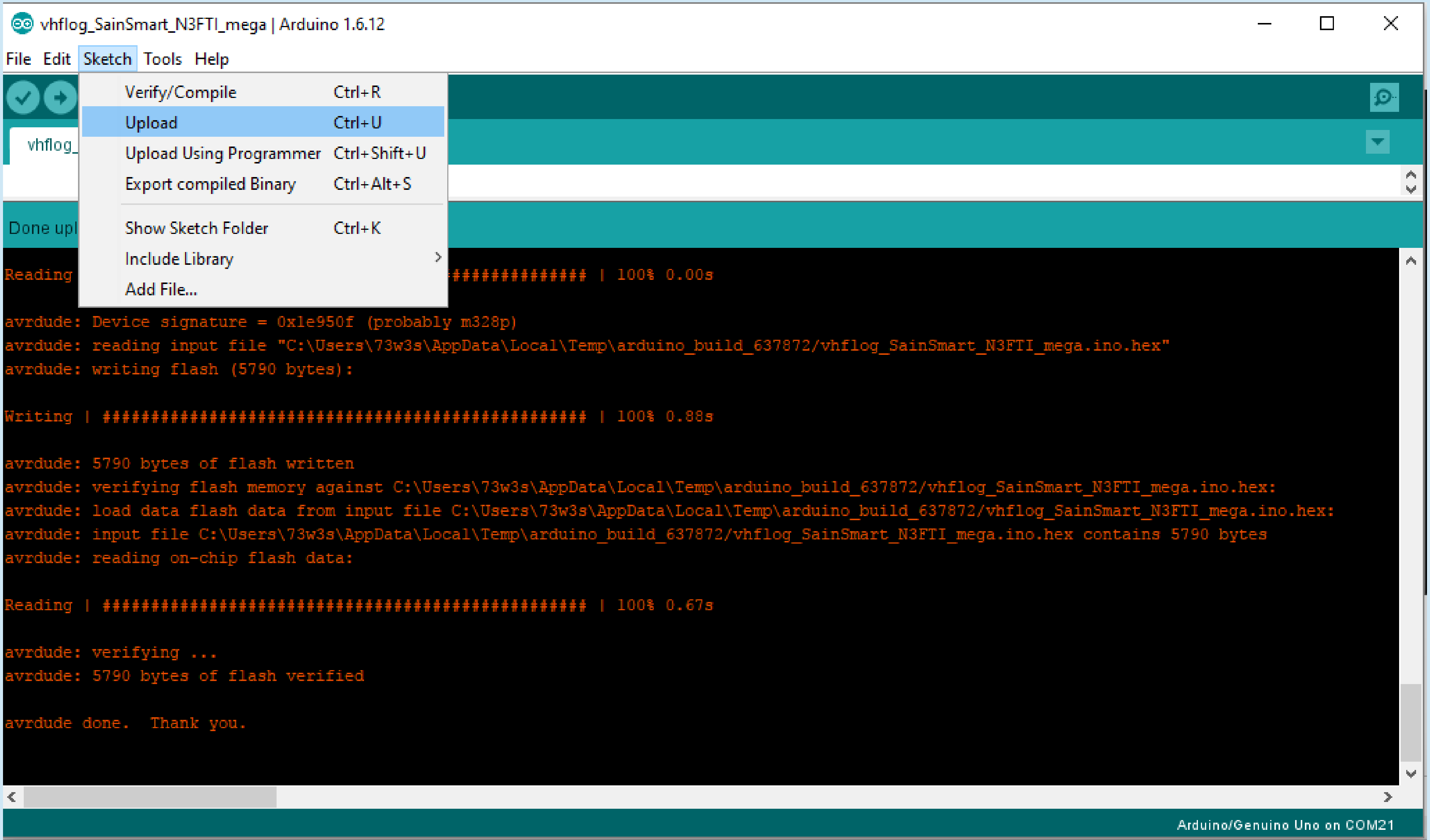
- Verify and Compile the code


vhflog_SainSmart_N3FTI_mega | Arduino 1.6.12
File Edit Sketch Tools Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
vhflog_
/*
Done cor
Arduino/Genuino Uno on COM21

```
"C:\Prog                                   \avr/bin/avr-gcc-ar" rcs  "C:\Users\73w3s\AppData\Local\Temp\arduino_build_637872\core\core.a" "C:
"C:\Prog                                   \avr/bin/avr-gcc-ar" rcs  "C:\Users\73w3s\AppData\Local\Temp\arduino_build_637872\core\core.a" "C:
"C:\Program Files (x86)\Arduino\hardware\tools\avr/bin/avr-gcc-ar" rcs  "C:\Users\73w3s\AppData\Local\Temp\arduino_build_637872\core\core.a" "C:
"C:\Program Files (x86)\Arduino\hardware\tools\avr/bin/avr-gcc-ar" rcs  "C:\Users\73w3s\AppData\Local\Temp\arduino_build_637872\core\core.a" "C:
"C:\Program Files (x86)\Arduino\hardware\tools\avr/bin/avr-gcc-ar" rcs  "C:\Users\73w3s\AppData\Local\Temp\arduino_build_637872\core\core.a" "C:
"C:\Program Files (x86)\Arduino\hardware\tools\avr/bin/avr-gcc-ar" rcs  "C:\Users\73w3s\AppData\Local\Temp\arduino_build_637872\core\core.a" "C:
"C:\Program Files (x86)\Arduino\hardware\tools\avr/bin/avr-gcc-ar" rcs  "C:\Users\73w3s\AppData\Local\Temp\arduino_build_637872\core\core.a" "C:
"C:\Program Files (x86)\Arduino\hardware\tools\avr/bin/avr-gcc-ar" rcs  "C:\Users\73w3s\AppData\Local\Temp\arduino_build_637872\core\core.a" "C:
"C:\Program Files (x86)\Arduino\hardware\tools\avr/bin/avr-gcc-ar" rcs  "C:\Users\73w3s\AppData\Local\Temp\arduino_build_637872\core\core.a" "C:
Linking everything together...
"C:\Program Files (x86)\Arduino\hardware\tools\avr/bin/avr-gcc" -w -Os -flto -fuse-linker-plugin -Wl,--gc-sections -mmcu=atmega328p  -o "C:\User
"C:\Program Files (x86)\Arduino\hardware\tools\avr/bin/avr-objcopy" -O ihex -j .eeprom --set-section-flags=.eeprom=alloc,load --no-change-warnin
"C:\Program Files (x86)\Arduino\hardware\tools\avr/bin/avr-objcopy" -O ihex -R .eeprom  "C:\Users\73w3s\AppData\Local\Temp\arduino_build_637872/

Sketch uses 5,790 bytes (17%) of program storage space. Maximum is 32,256 bytes.
Global variables use 264 bytes (12%) of dynamic memory, leaving 1,784 bytes for local variables. Maximum is 2,048 bytes.
```

Arduino
Getting Ready

- Upload the code


vhflog_SainSmart_N3FTI_mega | Arduino 1.6.12
File
Edit
Sketch
Tools
Help
Verify/Compile Ctrl+R
Upload Ctrl+U
Upload Using Programmer Ctrl+Shift+U
Export compiled Binary Ctrl+Alt+S
Show Sketch Folder Ctrl+K
Include Library
Add File...
vhflog_
Done upl
Arduino/Genuino Uno on COM21

```
Reading                        ############### | 100% 0.00s

avrdude: Device signature = 0x1e950f (probably m328p)
avrdude: reading input file "C:\Users\73w3s\AppData\Local\Temp\arduino_build_637872/vhflog_SainSmart_N3FTI_mega.ino.hex"
avrdude: writing flash (5790 bytes):

Writing | ################################################## | 100% 0.88s

avrdude: 5790 bytes of flash written
avrdude: verifying flash memory against C:\Users\73w3s\AppData\Local\Temp\arduino_build_637872/vhflog_SainSmart_N3FTI_mega.ino.hex:
avrdude: load data flash data from input file C:\Users\73w3s\AppData\Local\Temp\arduino_build_637872/vhflog_SainSmart_N3FTI_mega.ino.hex:
avrdude: input file C:\Users\73w3s\AppData\Local\Temp\arduino_build_637872/vhflog_SainSmart_N3FTI_mega.ino.hex contains 5790 bytes
avrdude: reading on-chip flash data:

Reading | ################################################## | 100% 0.67s

avrdude: verifying ...
avrdude: 5790 bytes of flash verified

avrdude done.  Thank you.
```

Programming – A Few Terms

- **Comment**
  - Statement that is used to make a program easier to understand, and which is ignored by the computer
    - Start with // in C, or with # in Python, or ‘ in Basic
- **Data type**
  - Specifies the type of value for a variable or constant
    - String
      - “This is a string”
    - char(acter)
      - ‘g’
    - int(eger)
      - 3
    - Bool(ean)
      - True
    - Double or Float
      - 3.78943236593

Programming – A Few Terms

- **Library**
  - a collection of precompiled routines that a program can use
    - Ethernet.h
- **Variable**
  - a storage location paired with an associated symbolic name (an identifier), which contains some known or unknown quantity of information referred to as a value.
    - double myAirspeed;
- **Constant**
  - a value that cannot be altered by the program during normal execution
    - const double MyPi = 3.14159;
- **GPIO pin**
  - generic pin on an integrated circuit or computer board whose behavior—including whether it is an input or output pin—is controllable by the user at run time
    - GPIO.setup(PIN50, GPIO.OUT);

Programming – A Few Terms

- **Analog pin**
  - Pin that can read (or sometimes write) analog voltages within a defined range with step size determined by the bit size of the analog-to-digital (read) or digital-to-analog (write) converter
    - pinMode(A0, INPUT);
- **Statement**
  - the smallest standalone element of a programming language that expresses some action to be carried out. In C, every statement ends with a semi-colon
    - X = 1;

Programming – A Few Terms

Argument

Result

- **Function**
  - a named section of a program that performs a specific task and returns a value
    - int X (int y) { int x = y + 5; return x; }
  - “X” would be used in this manner:
    - int A = X(7);
      - Would give A = 12
  - A function’s “type” represents the data type that it returns when called

Programming – A Few Terms

Argument

- **Procedure**
  - a named section of a program that performs a specific task (such as I/O) but doesn't return a result. Example:

    ```
    void calcA (int y) { A = y + 5; }
    ```

    (where "A" is declared elsewhere in the program)
  - "calcA" would be used in this manner:

    ```
    Int A = 0;
    CalcA(7);
    Serial.print(A);
    ```

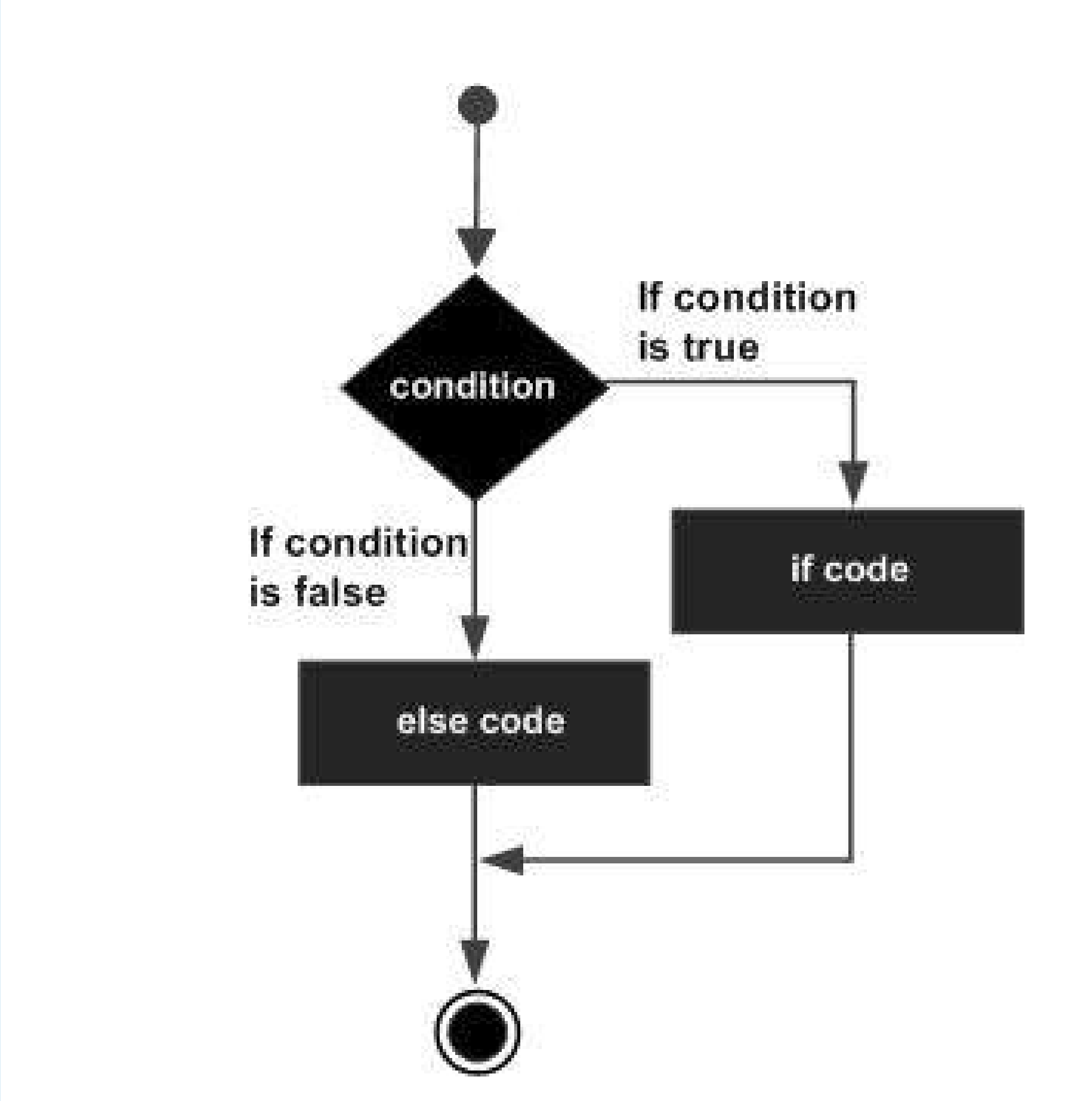
    - Would print "12" to the serial port
  - **A procedure's type is always "void"**
  - Another example:

    Serial.begin(9600); ARDUINO OFFICIALLY CALLS THIS A "FUNCTION"

Programming – A Few Terms

- **Operator**
  - A character that represents an action
    - +, -, *, /
- **Decision Statement**
  - Statement that causes various courses of action to be taken depending on certain conditions
  - **if, else if, else** (if, elif, else in Python)
    - if
    - if...else
    - if...else if
    - if...else if...else
    - if...else if...else if
    - if...else if...else if...else   etc.
  - **switch**

if/else Statement

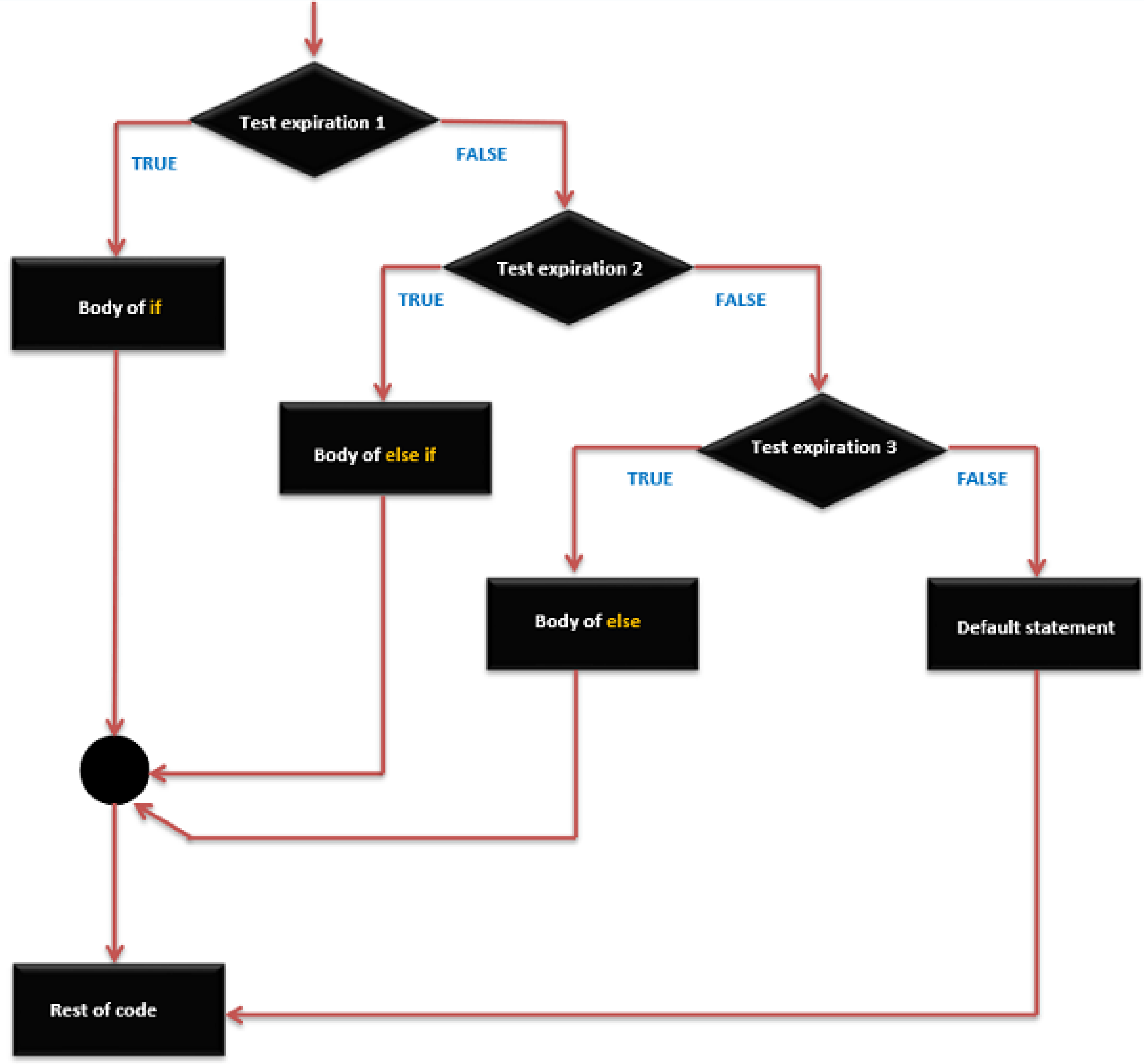


if / else if / else
C

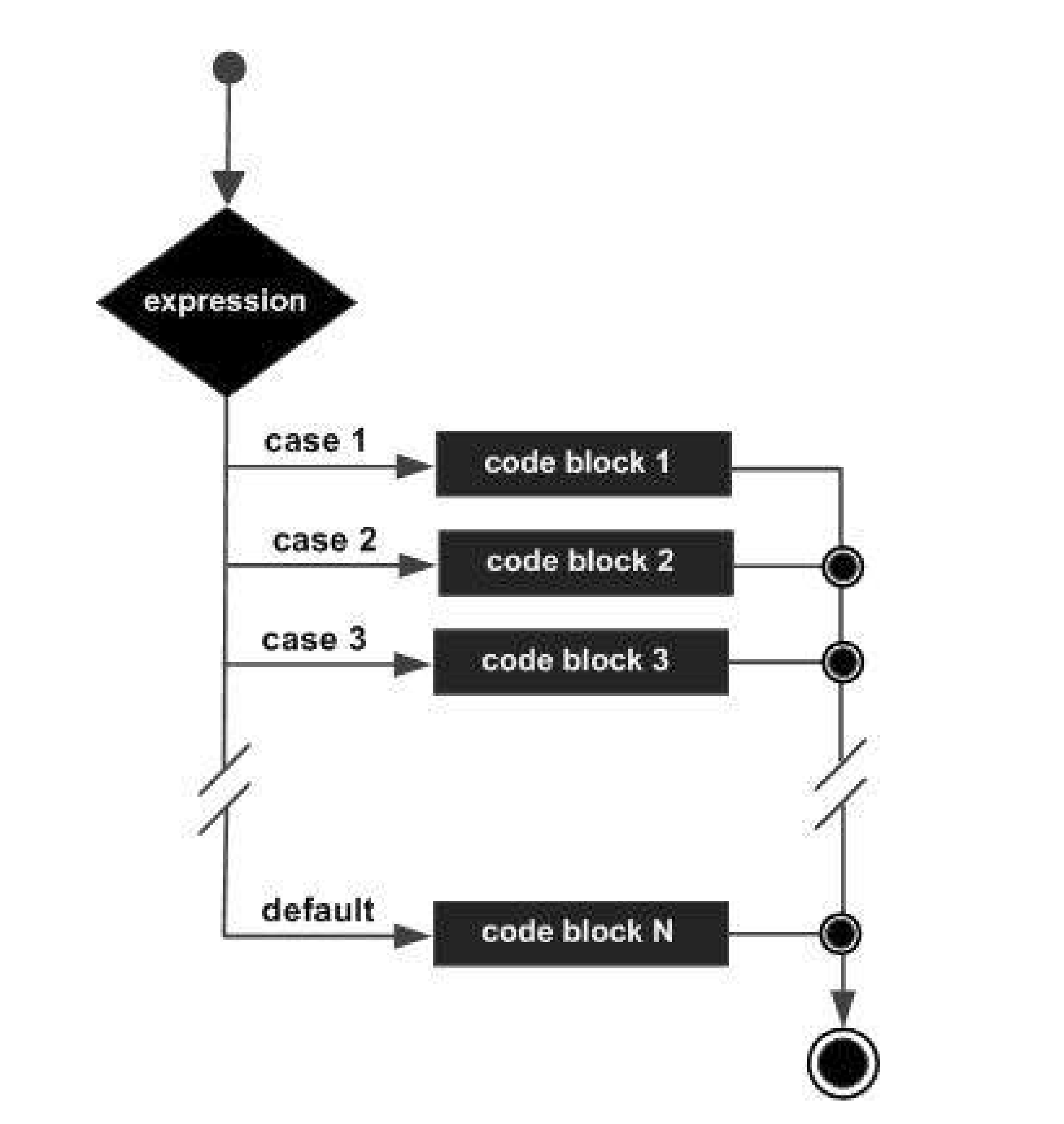
```
if (band == 50)
{relay50Pin = On;
relay144Pin = Off;}
else if (band ==144)
{relay50Pin = Off;
relay144Pin = On;}
else
{relay50Pin = Off;
relay144Pin = Off;}
```

if / elif / else
python

```
if band == 50:
    relay50Pin = On
    relay144Pin = Off
elif band ==144:
    relay50Pin = Off
    relay144Pin = On
else:
    relay50Pin = Off
    relay144Pin = Off
```


Test expiration 1
TRUE
FALSE
Body of if
Test expiration 2
TRUE
FALSE
Body of else if
Test expiration 3
TRUE
FALSE
Body of else
Default statement
Rest of code

switch : an alternative to if / else if / else if / else if …. / else



FOR INTEGRAL and ENUMERATED TYPES ONLY e.g. integers, characters, booleans

switch

```
• switch (band){
    case 50: {
        relay50Pin = On;
        relay144Pin = Off;
        break; }
    case 144: {
        relay50Pin = Off;
        relay144Pin = On;
        break; }
    default: {
        relay50Pin = Off;
        relay144Pin = Off; }}
```

python DOES NOT HAVE A DIRECT EQUIVALENT

Curly braces

- Used to group a set of statements; always come in pairs
- Not used in python, where indentation defines groups of statements
- Be careful with them; misplaced curley braces are a major source of bugs!

Curly brace madness

- if (band == 50)
  {relay50Pin = On;
  relay144Pin = Off;}
  else if (band ==144)
  {relay50Pin = Off;
  relay144Pin = On;}
  else
  {relay50Pin = Off;
  relay144Pin = Off;}

- if (band == 50)
  {relay50Pin = On;
  relay144Pin = Off;}
  else if (band ==144)
  {relay50Pin = Off;
  relay144Pin = On;}
  else
  {relay50Pin = Off;}
  relay144Pin = Off;

Shortcuts

- +=

    A += 2;

    is the same as

    A = A + 2;

- *=

    A *= 2;

    is the same as

    A = A * 2;

- Same for subtraction, division, exponentiation, etc.

= vs ==

- = is an assignment operator:

```
A = B + 5;
```

- == is a comparison operator (the “Equal To” operator):

```
if (A ==B)
    {print (“A equals B”);}
else
    {print (“They are not the same”);}
```

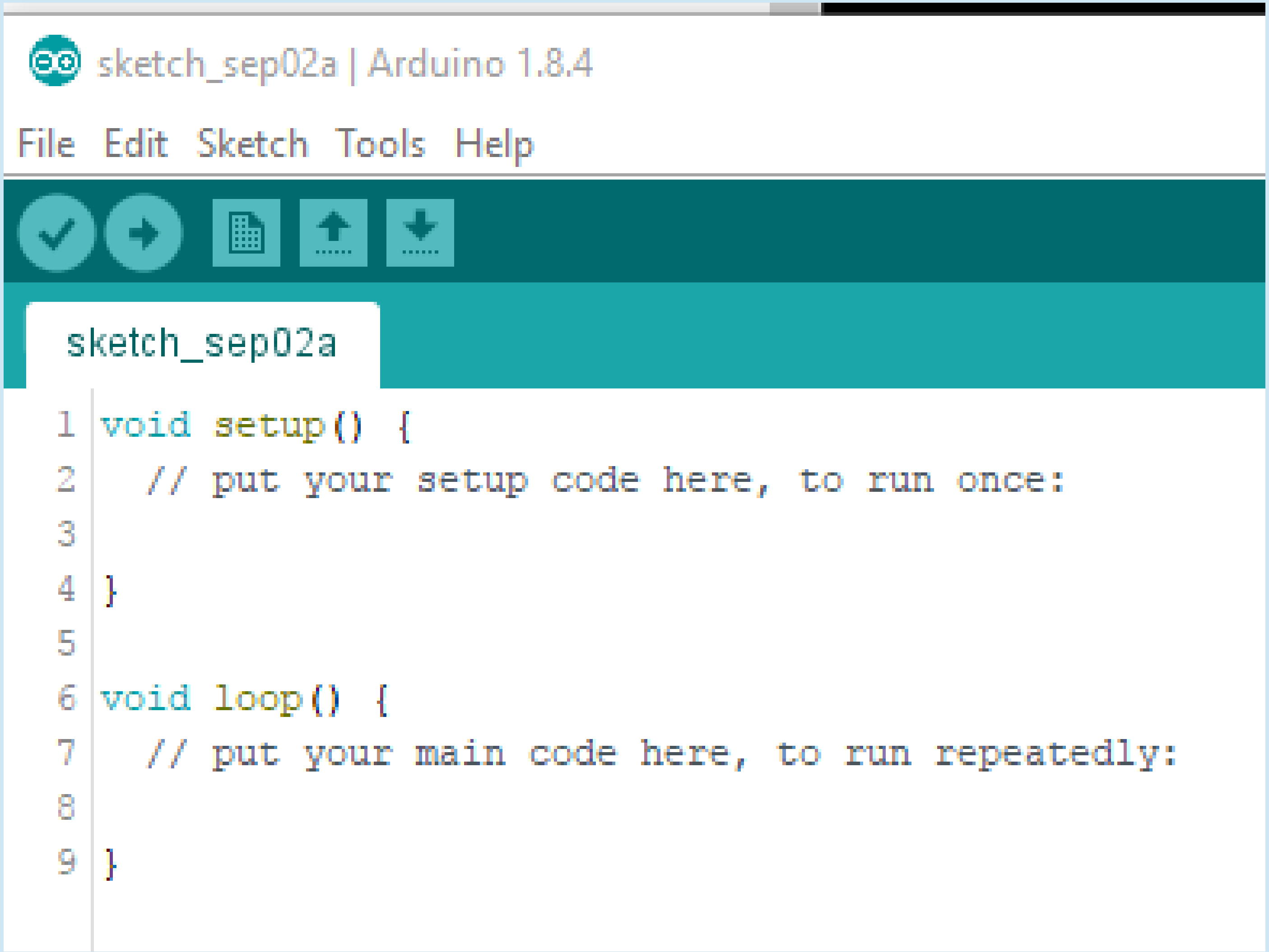
!

- ! is the Logical NOT operator
  - It means “is not”

    Example:

    ```
    if (A != 2)
        {print(“A is not equal to 2”);}
    else
        {print(“A is equal to 2”);}
    ```

Programming Steps – Getting Started



```
void setup() {
  // put your setup code here, to run once:

}

void loop() {
  // put your main code here, to run repeatedly:

}
```

Programming Steps - General

1) Include libraries containing external functions
2) Define variables and constants
3) Setup ()

   Define and initialize GPIO pins / Analog I/O pins

   Define, start, serial port(s), Ethernet port(s)

4) Loop()

   Receive input from ports / GPIO pins / Analog pins

   Parse / process data to extract desired information

   Use information derived from data to perform desired task (e.g. switch GPIO pins) or to send information to client computer

5) From within Loop(), call other functions() as needed

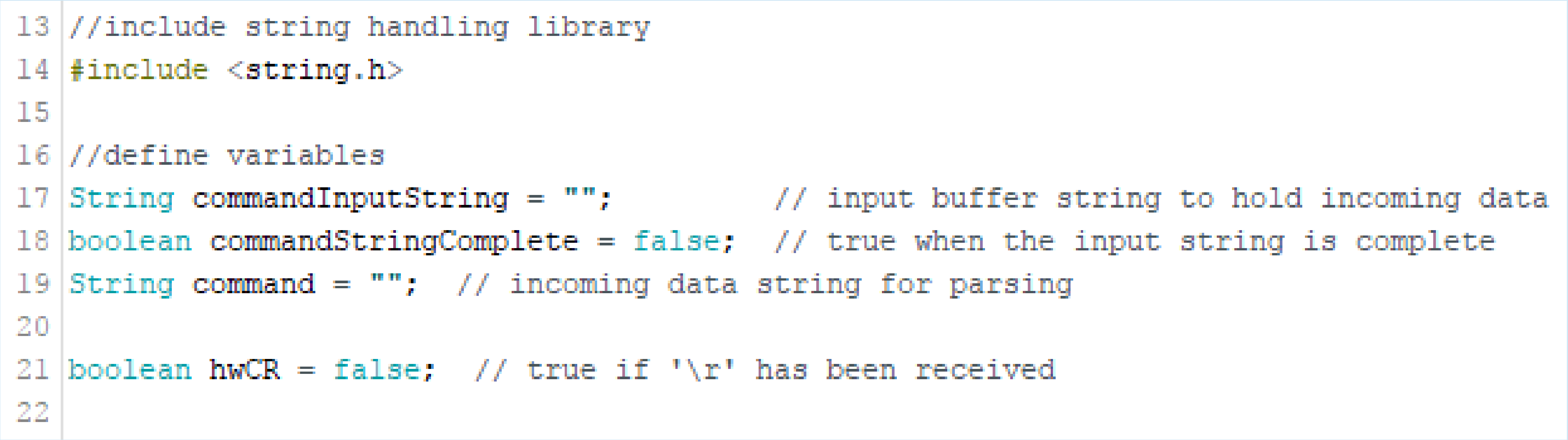
Programming Steps -
Arduino USB-Serial Example

1) Include libraries containing external functions
2) Define variables and constants
3) Setup()
    - Define and initialize GPIO pins
    - Define, start, flush serial port
4) Loop()
    1) Receive serial data sent by logger (Function serialEvent())
    2) Parse serial data from logger and extract band information
    3) Use band information to switch bands using GPIO Pins

See pages 2-8 in Code Handout

Arduino Example

Include Libraries and Define Variables

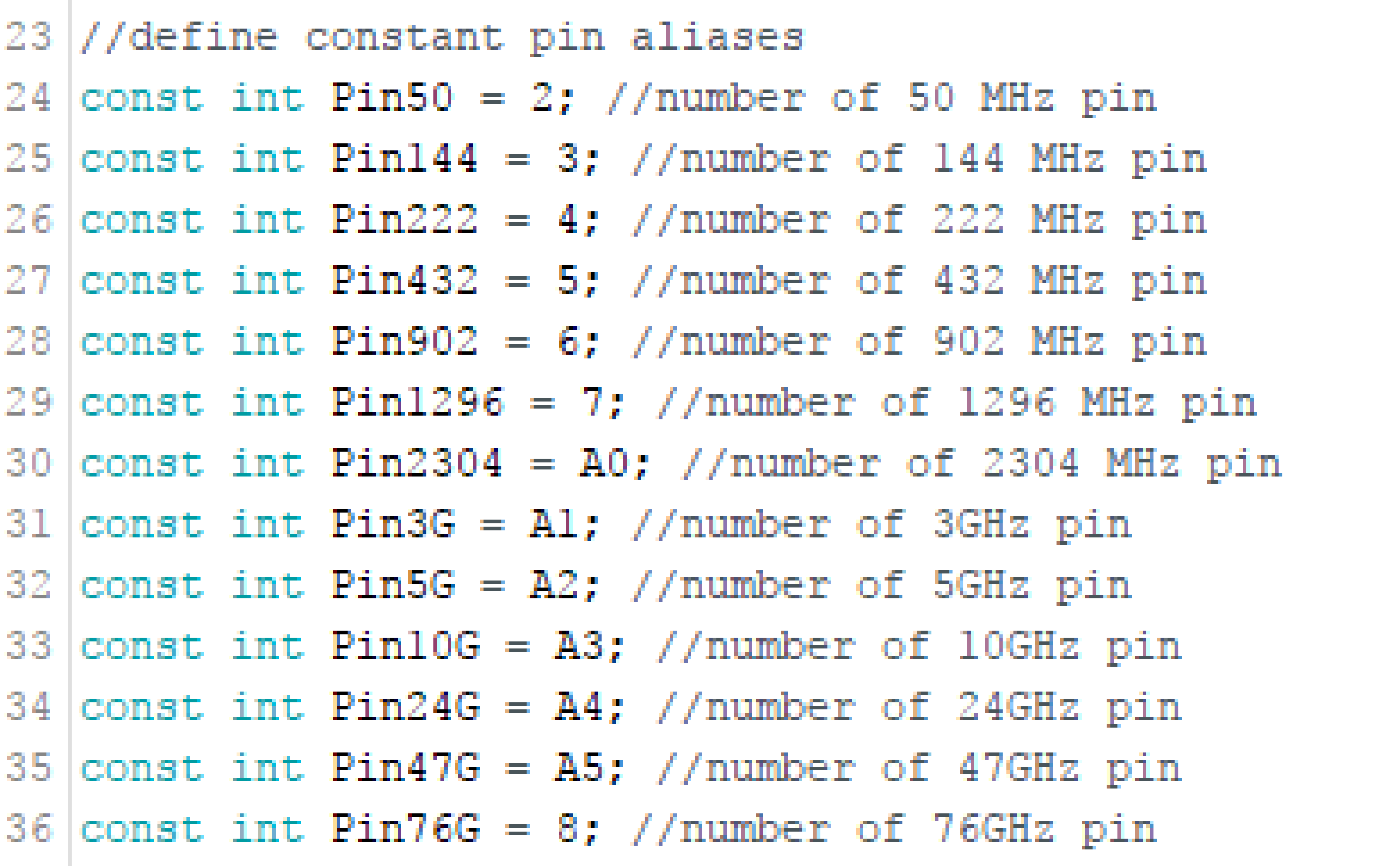


```
//include string handling library
#include <string.h>

//define variables
String commandInputString = "";        // input buffer string to hold incoming data
boolean commandStringComplete = false;  // true when the input string is complete
String command = "";  // incoming data string for parsing

boolean hwCR = false;  // true if '\r' has been received

```

Page 3 Code Handout

Arduino Example

Define Constants (GPIO Pin Aliases are constants)

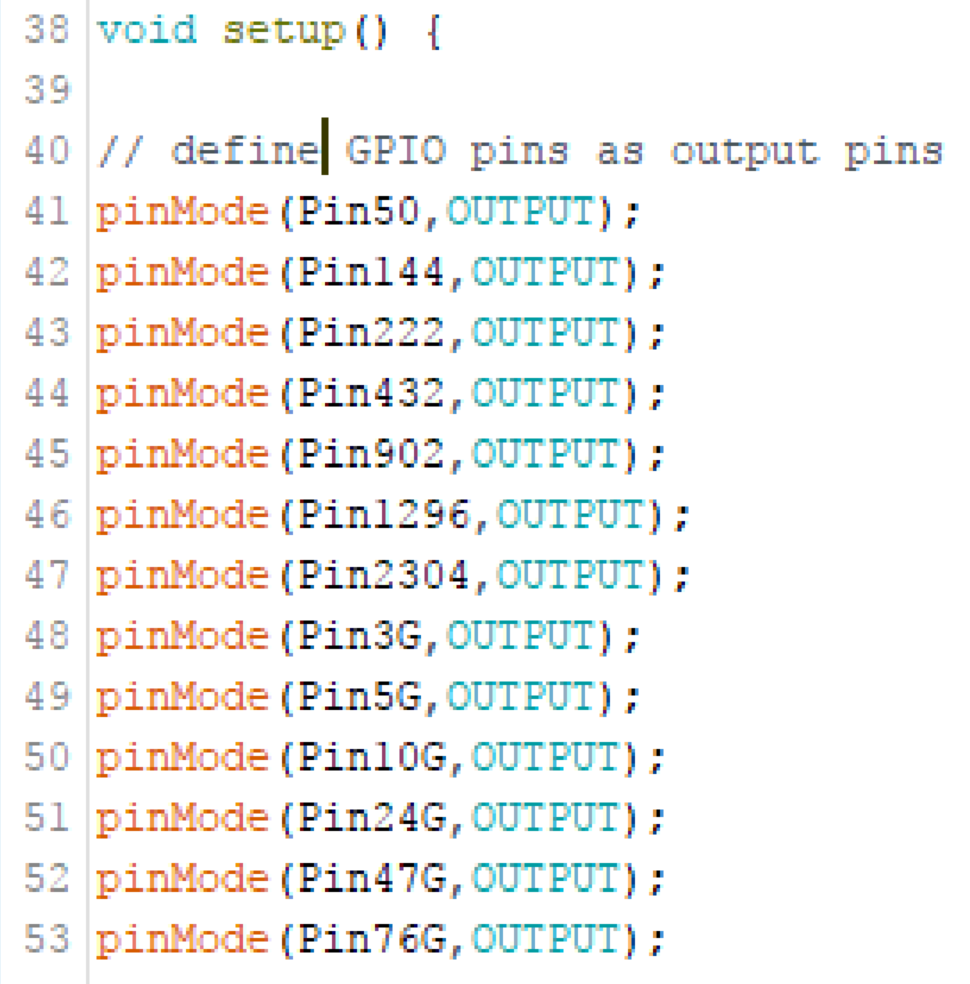


```
//define constant pin aliases
const int Pin50 = 2; //number of 50 MHz pin
const int Pin144 = 3; //number of 144 MHz pin
const int Pin222 = 4; //number of 222 MHz pin
const int Pin432 = 5; //number of 432 MHz pin
const int Pin902 = 6; //number of 902 MHz pin
const int Pin1296 = 7; //number of 1296 MHz pin
const int Pin2304 = A0; //number of 2304 MHz pin
const int Pin3G = A1; //number of 3GHz pin
const int Pin5G = A2; //number of 5GHz pin
const int Pin10G = A3; //number of 10GHz pin
const int Pin24G = A4; //number of 24GHz pin
const int Pin47G = A5; //number of 47GHz pin
const int Pin76G = 8; //number of 76GHz pin
```

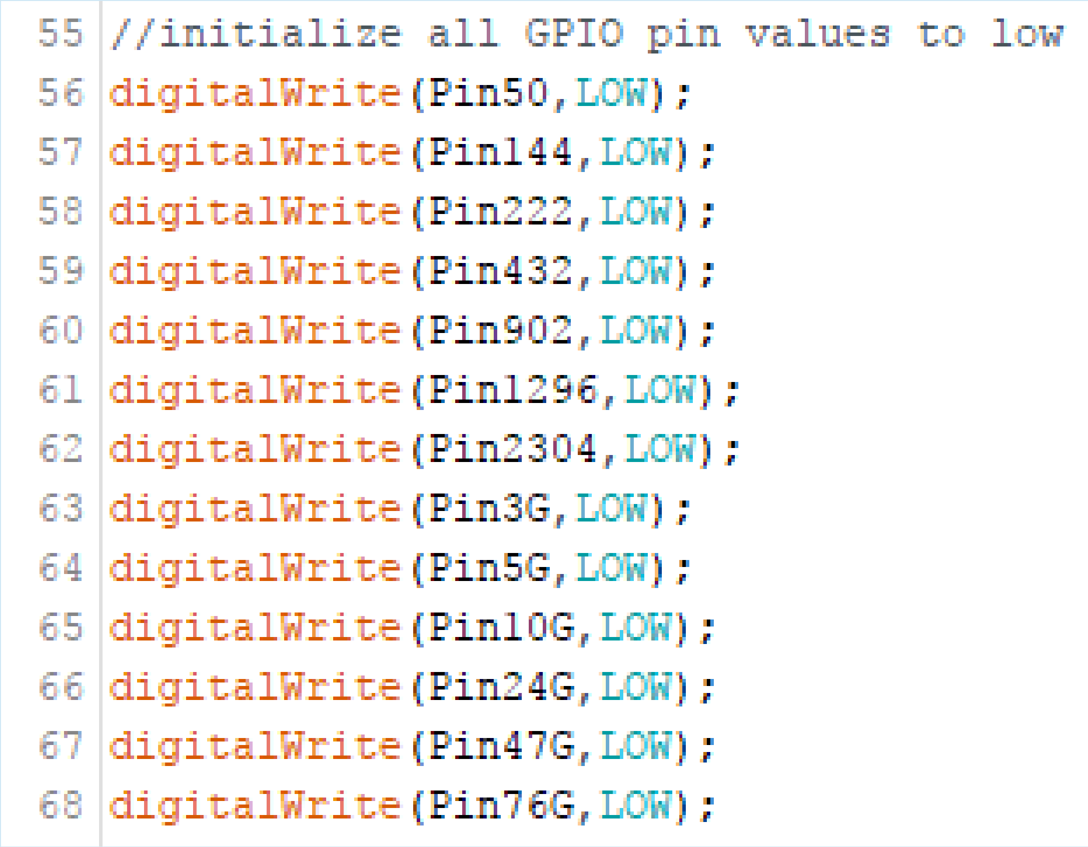
Page 3 Code Handout

Arduino Example

Define and Initialize GPIO Pins



```
void setup() {

// define GPIO pins as output pins
pinMode(Pin50,OUTPUT);
pinMode(Pin144,OUTPUT);
pinMode(Pin222,OUTPUT);
pinMode(Pin432,OUTPUT);
pinMode(Pin902,OUTPUT);
pinMode(Pin1296,OUTPUT);
pinMode(Pin2304,OUTPUT);
pinMode(Pin3G,OUTPUT);
pinMode(Pin5G,OUTPUT);
pinMode(Pin10G,OUTPUT);
pinMode(Pin24G,OUTPUT);
pinMode(Pin47G,OUTPUT);
pinMode(Pin76G,OUTPUT);
```

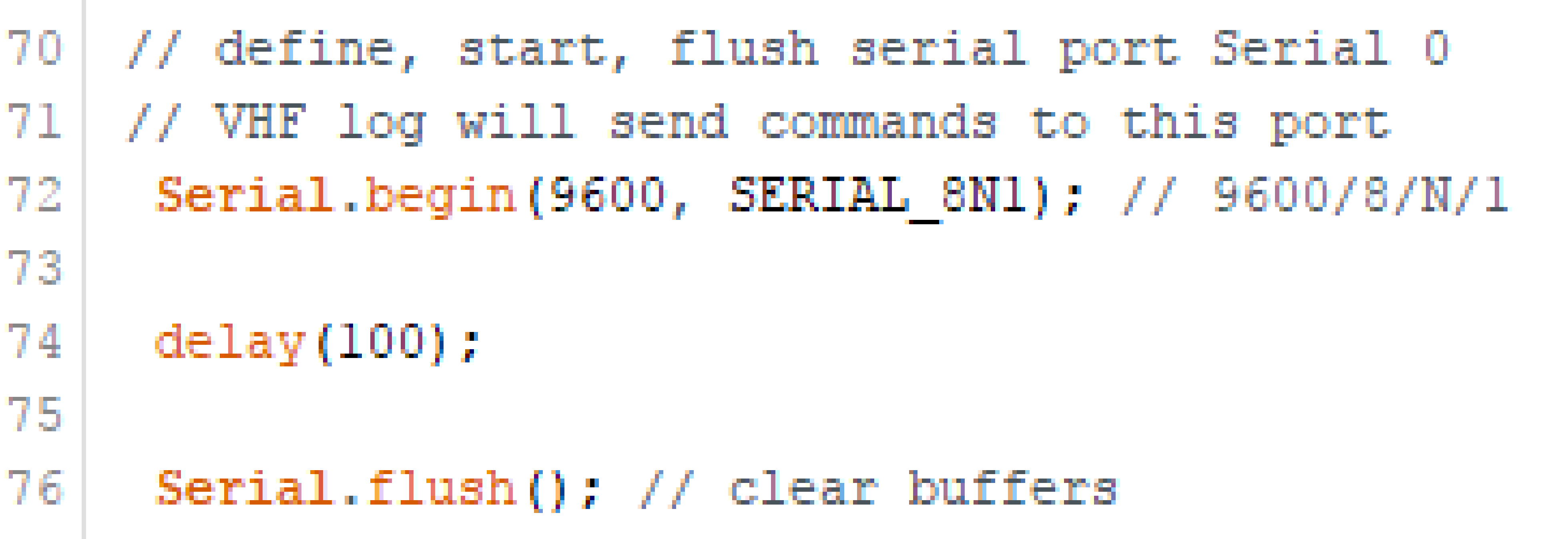


```
//initialize all GPIO pin values to low
digitalWrite(Pin50,LOW);
digitalWrite(Pin144,LOW);
digitalWrite(Pin222,LOW);
digitalWrite(Pin432,LOW);
digitalWrite(Pin902,LOW);
digitalWrite(Pin1296,LOW);
digitalWrite(Pin2304,LOW);
digitalWrite(Pin3G,LOW);
digitalWrite(Pin5G,LOW);
digitalWrite(Pin10G,LOW);
digitalWrite(Pin24G,LOW);
digitalWrite(Pin47G,LOW);
digitalWrite(Pin76G,LOW);
```

Page 3 Code Handout

Arduino Example

Define, Start, Flush Serial Port

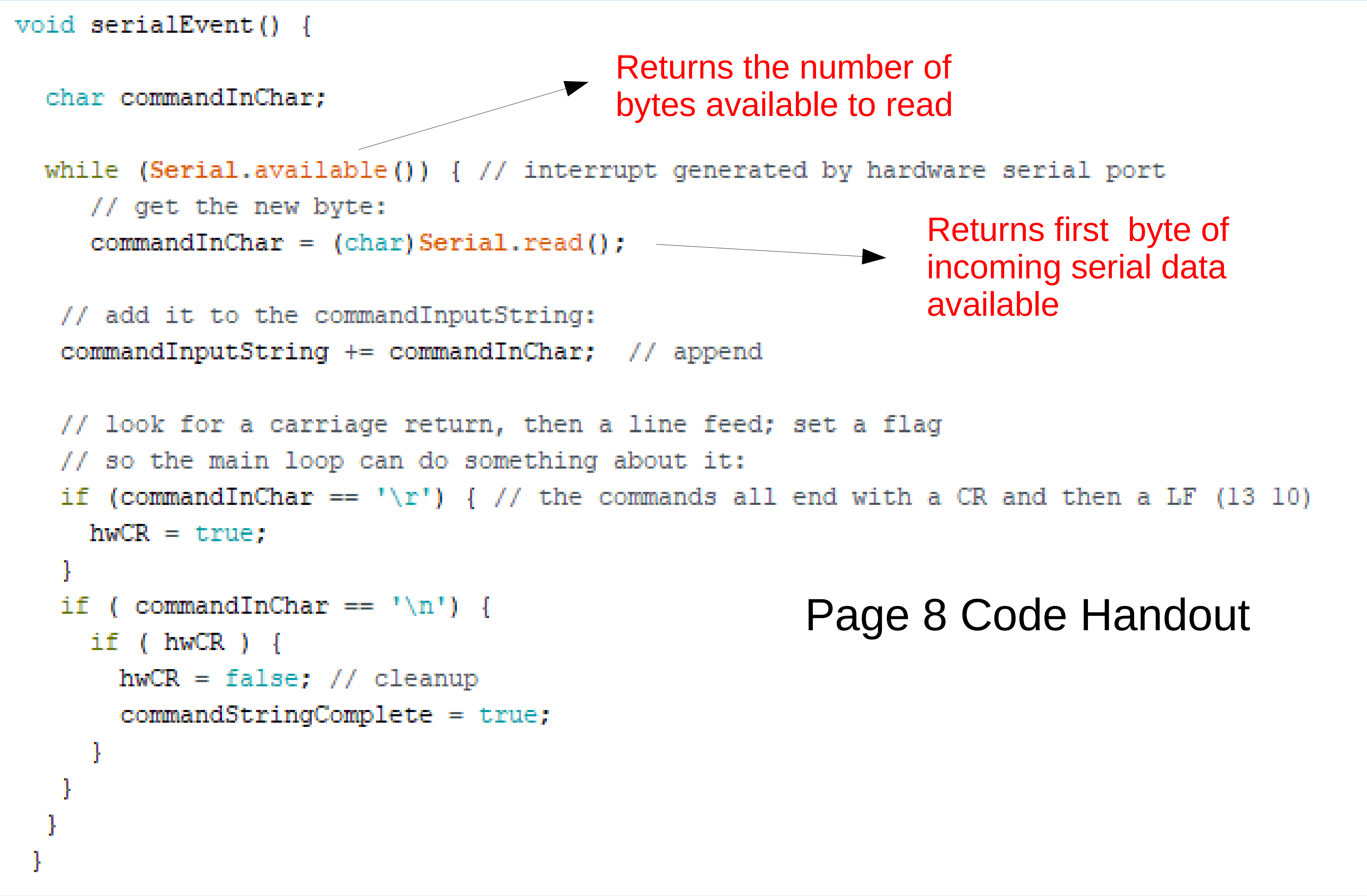


```
  // define, start, flush serial port Serial 0
  // VHF log will send commands to this port
   Serial.begin(9600, SERIAL_8N1); // 9600/8/N/1

   delay(100);

   Serial.flush(); // clear buffers
```

Page 4 Code Handout

Arduino Example

Receive serial data sent by logger



```
void serialEvent() {

  char commandInChar;

  while (Serial.available()) { // interrupt generated by hardware serial port
     // get the new byte:
     commandInChar = (char)Serial.read();

   // add it to the commandInputString:
   commandInputString += commandInChar;  // append

   // look for a carriage return, then a line feed; set a flag
   // so the main loop can do something about it:
   if (commandInChar == '\r') { // the commands all end with a CR and then a LF (13 10)
     hwCR = true;
   }
   if ( commandInChar == '\n') {
     if ( hwCR ) {
       hwCR = false; // cleanup
       commandStringComplete = true;
     }
   }
  }
 }
```

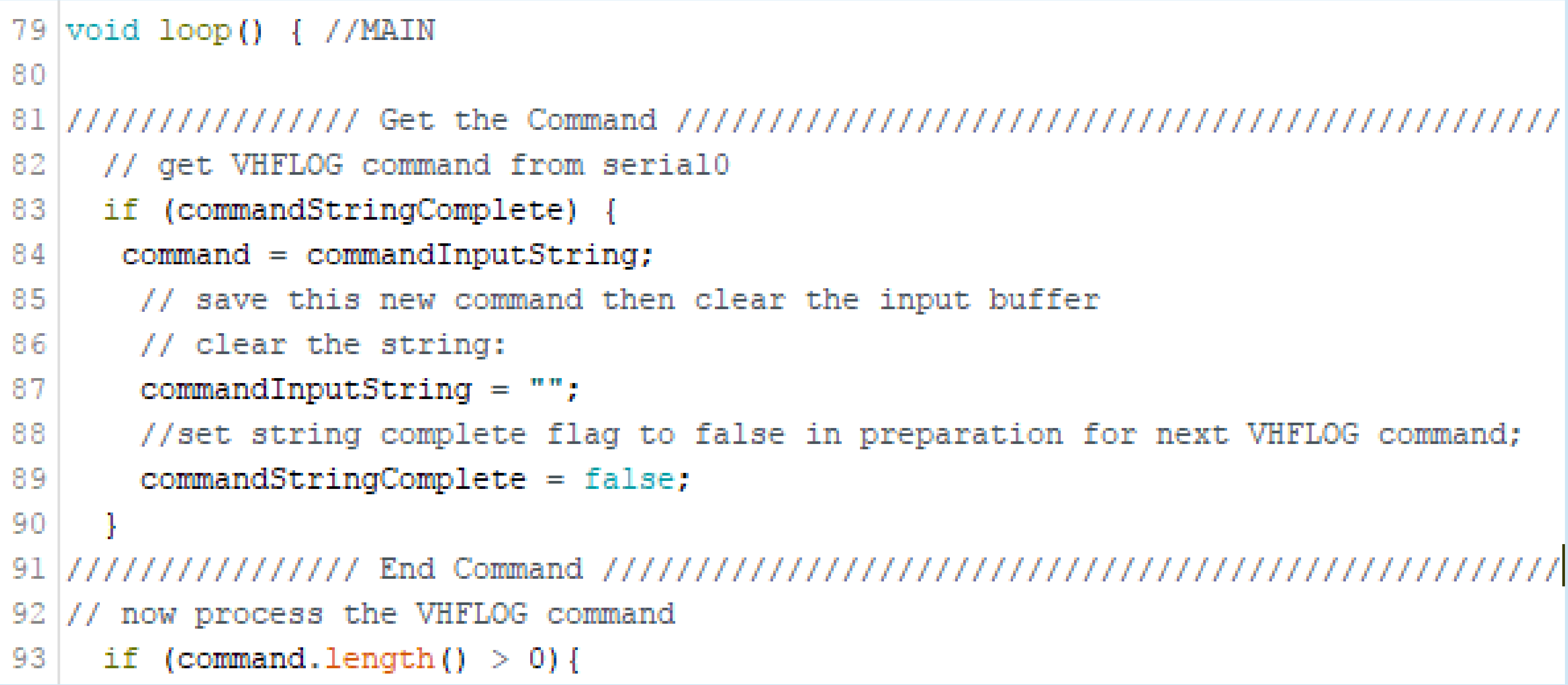
Returns the number of bytes available to read

Returns first byte of incoming serial data available

Page 8 Code Handout

Arduino Example

Main Loop – Waits for Serial Data
If Serial Data Present, Starts Parsing

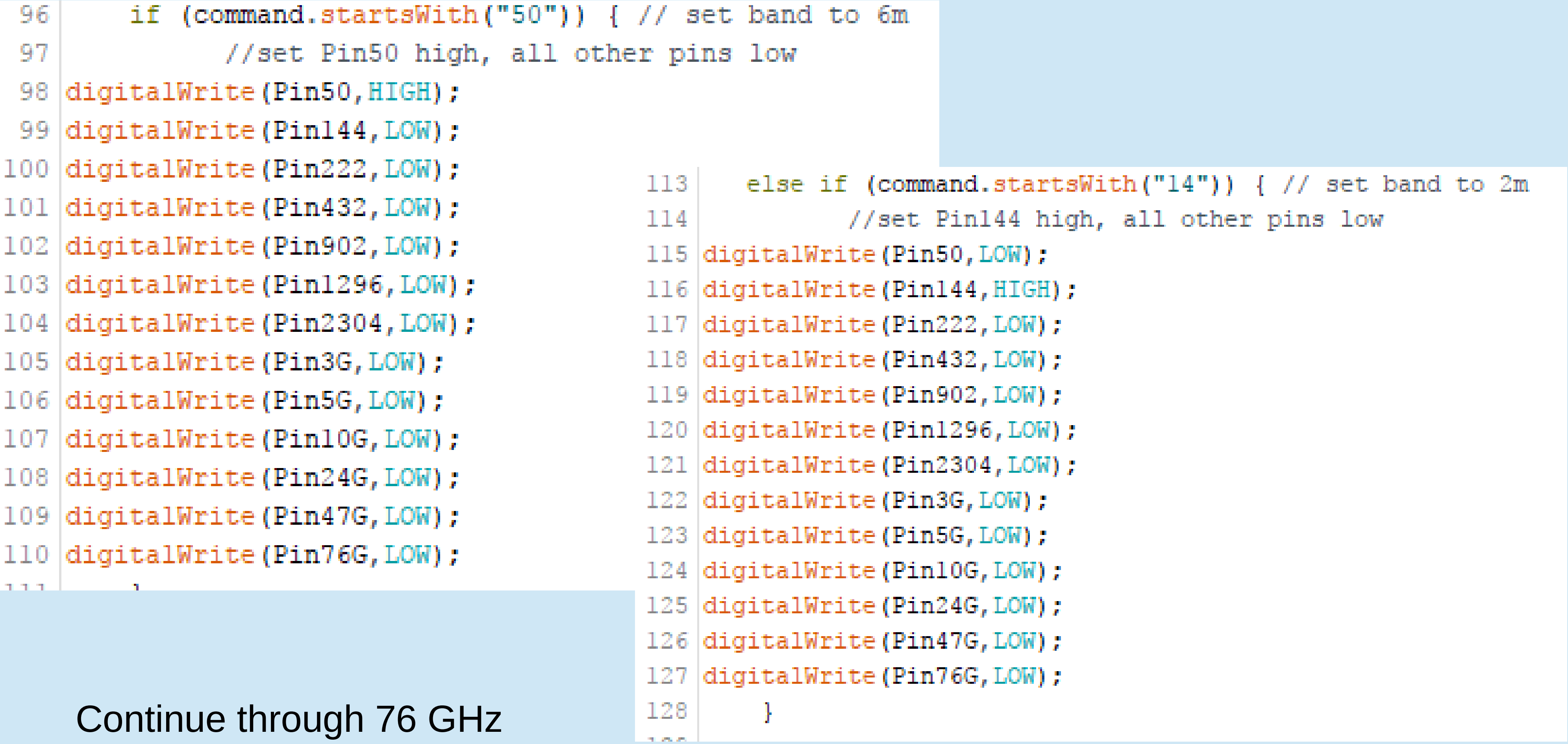


```
void loop() { //MAIN

//////////////// Get the Command ////////////////////////////////////////////////
  // get VHFLOG command from serial0
  if (commandStringComplete) {
   command = commandInputString;
    // save this new command then clear the input buffer
    // clear the string:
    commandInputString = "";
    //set string complete flag to false in preparation for next VHFLOG command;
    commandStringComplete = false;
  }
//////////////// End Command ////////////////////////////////////////////////////
// now process the VHFLOG command
  if (command.length() > 0){
```

Page 4 Code Handout

Arduino Example

Parse Serial Data, Extract Band Info, Bandswitch



```
    if (command.startsWith("50")) { // set band to 6m
        //set Pin50 high, all other pins low
digitalWrite(Pin50,HIGH);
digitalWrite(Pin144,LOW);
digitalWrite(Pin222,LOW);
digitalWrite(Pin432,LOW);
digitalWrite(Pin902,LOW);
digitalWrite(Pin1296,LOW);
digitalWrite(Pin2304,LOW);
digitalWrite(Pin3G,LOW);
digitalWrite(Pin5G,LOW);
digitalWrite(Pin10G,LOW);
digitalWrite(Pin24G,LOW);
digitalWrite(Pin47G,LOW);
digitalWrite(Pin76G,LOW);
```

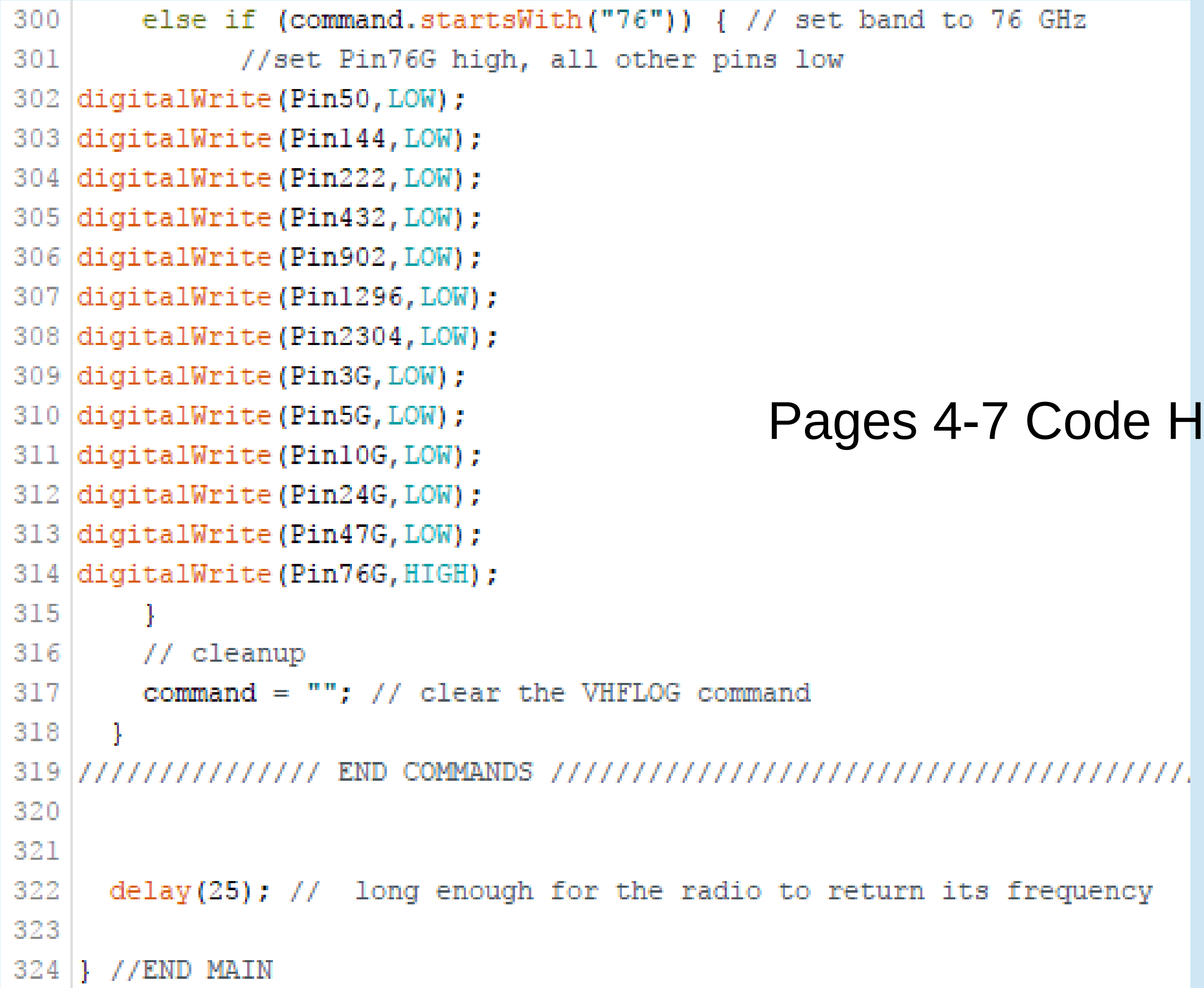
```
   else if (command.startsWith("14")) { // set band to 2m
        //set Pin144 high, all other pins low
digitalWrite(Pin50,LOW);
digitalWrite(Pin144,HIGH);
digitalWrite(Pin222,LOW);
digitalWrite(Pin432,LOW);
digitalWrite(Pin902,LOW);
digitalWrite(Pin1296,LOW);
digitalWrite(Pin2304,LOW);
digitalWrite(Pin3G,LOW);
digitalWrite(Pin5G,LOW);
digitalWrite(Pin10G,LOW);
digitalWrite(Pin24G,LOW);
digitalWrite(Pin47G,LOW);
digitalWrite(Pin76G,LOW);
    }
```

Continue through 76 GHz

Pages 4-7 Code Handout

Arduino Example

Finish Parsing, Bandswitching, Clean Up



```
    else if (command.startsWith("76")) { // set band to 76 GHz
          //set Pin76G high, all other pins low
digitalWrite(Pin50,LOW);
digitalWrite(Pin144,LOW);
digitalWrite(Pin222,LOW);
digitalWrite(Pin432,LOW);
digitalWrite(Pin902,LOW);
digitalWrite(Pin1296,LOW);
digitalWrite(Pin2304,LOW);
digitalWrite(Pin3G,LOW);
digitalWrite(Pin5G,LOW);
digitalWrite(Pin10G,LOW);
digitalWrite(Pin24G,LOW);
digitalWrite(Pin47G,LOW);
digitalWrite(Pin76G,HIGH);
    }
    // cleanup
    command = ""; // clear the VHFLOG command
  }
//////////////// END COMMANDS //////////////////////////////////////////


  delay(25); //  long enough for the radio to return its frequency

} //END MAIN
```

Pages 4-7 Code Handout

Station Automation Coding

- Very Simple:

  Got Some Input

  Did Something With It

  Produced Some Output

Programming Steps - Arduino USB-Serial Example

1) Included libraries containing external functions:

   string.h

2) Defined variables and constants

3) Setup()

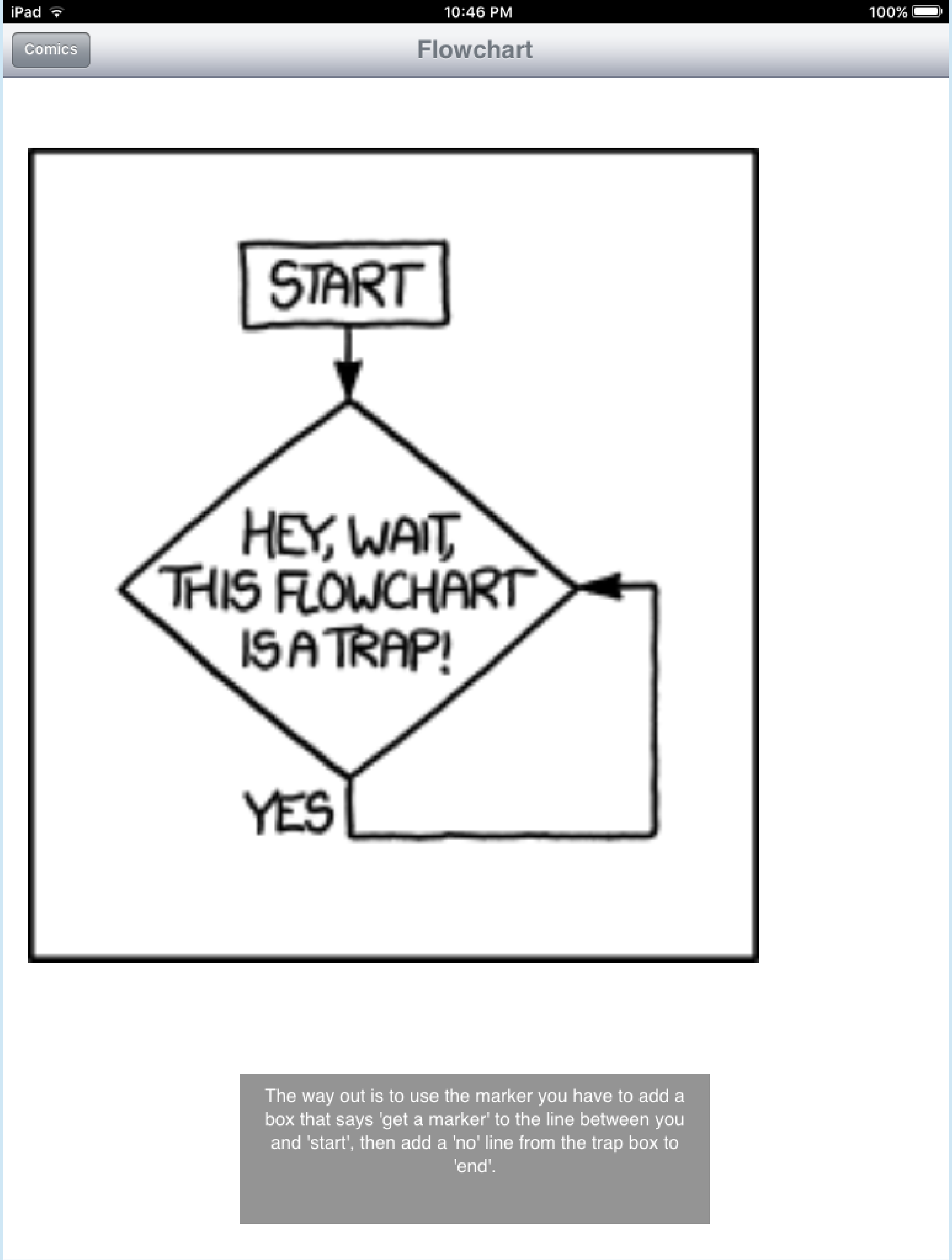
   Defined and initialized GPIO pins

   Defined, started, flushed serial port

4) Loop()

   1) Received serial data sent by logger (Function serialEvent())
   2) Parsed serial data from logger and extracted band information
   3) Used band information to switch bands using GPIO Pin Outputs

See pages 2-8 in Code Handout



The way out is to use the marker you have to add a box that says 'get a marker' to the line between you and 'start', then add a 'no' line from the trap box to 'end'.