Assignment 1.1

Steinarr Hrafn Höskuldsson

August 18, 2022

Part 1

By commenting and uncommenting the declarations we can get the following information on RAM usage:

- default int gives: 97.7% (used 2000 bytes from 2048 bytes)
- 8 bit int gives: 48.8% (used 1000 bytes from 2048 bytes)
- 16 bit int gives 97.7% (used 2000 bytes from 2048 bytes)
- 32 bit int gives 195.3 % (used 4000 bytes from 2048 bytes)
- 64 bit int gives 390.6% (used 8000 bytes from 2048 bytes)

The 32 and 64 bit tests exceeded the available RAM. The 16 bit test did not but leaves almost no space for local variables. We can now infer that the Arduino uses 16 bit integers as the default.

Part 2

Global

We saw the effect of having a global variable in Part 1. The 2000 bytes are stored in RAM at compile time

Local

Declaring a as a local variable has no effect on the RAM at compile time. We can expect it to take up RAM space at runtime.

Static

Declaring a as a local static variable does take up space in RAM at compile time. 2000 bytes as before.

Constant

Declaring a as a local constant does not take up space in RAM nor does it take up space in Flash. This makes sense, if, at runtime, someone needs to know what the value of anything in a is, it is 0. Only a few bytes of Flash needed to remember that. If we were to initialize a with some values. Those values would need to be stored and thus take up space in Flash.

Dynamic

Allocating a as a dynamic appears to take up approximately 750 bytes of Flash memory but on further testing by adding more and more dynamic variables the Flash memory requirement becomes smaller and smaller. Thus I draw the conclusion that the bytes needed in Flash memory actually come from including the instruction sets needed for dynamic memory allocation.

Code

Mesaurements were taken by uncommenting the decleration of interest at each time.

```
#include <stdint.h> // Defines integer types of a particular size

////////   PART 1   ///////////
// default int:
int a[1000];  // 97.7% (used 2000 bytes from 2048 bytes)

// 8 bit int:
// uint8_t a[1000]; // 48.8% (used 1000 bytes from 2048 bytes)

// 16 bit int:
// uint16_t a[1000];  // 97.7% (used 2000 bytes from 2048 bytes)

// 32 bit int:
// uint32_t a[1000];  // 195.3 % (used 4000 bytes from 2048 bytes)

// 64 bit int:
// uint64_t a[1000]; // 390.6% (used 8000 bytes from 2048 bytes)

///////   PART 2   /////////////////
// global
// int a[10000];

// global constant
// const int a[10000];

int main()
{
  // local
  // int a[1000];

  // local static
  // static int a[1000];

  // local constant
  // const int a[1000];

  // dynamic variable
  // int *a = new int[1000];
  // int *b = new int[1000];
  // int *c = new int[1000];
  // int *d = new int[1000];

  return a[0] ;  //  + b[0] + c[0] + d[0];
}
```