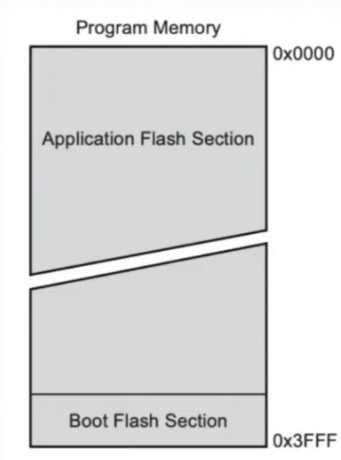
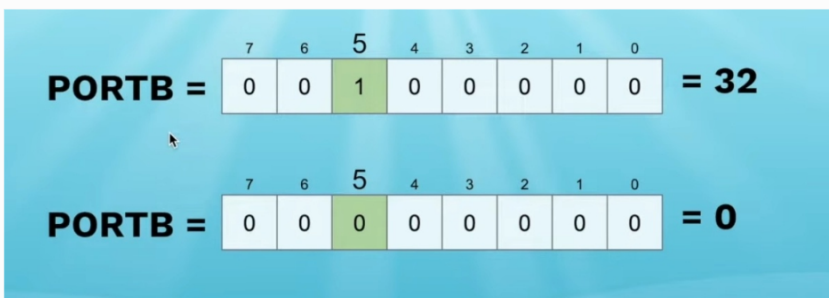


Flash Memory



• built-in Led (D13) (PB5) (port B, bit 5)



```
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  digitalWrite(LED_BUILTIN, HIGH);  
  delay(1000);  
  digitalWrite(LED_BUILTIN, LOW);  
  delay(1000);  
}
```

```
void setup() {  
  // initialize digital pin LED_BUILTIN as an output.  
  pinMode(LED_BUILTIN, OUTPUT);  
}  
  
void loop() {  
  PORTB = 32; // 00100000  
  delay(1000);  
  PORTB = 0; // 00000000  
  delay(1000);  
}
```

```
void loop() {
  PORTB = 32;
  PORTB = 0;
}
```

18.7x faster!

```
void loop() {
  digitalWrite(13,HIGH);
  digitalWrite(13,LOW);
}
```

2.78 MHz

148 kHz

```
void setup() {
  pinMode(LED_BUILTIN, OUTPUT);
}
```

```
void loop() {
  PORTB = 32;
  delay(1000);
  PORTB = 0;
  delay(1000);
}
```

13.4.2 PORTB – The Port B Data Register

Bit	7	6	5	4	3	2	1	0	
0x05 (0x25)	PORTB7	PORTB6	PORTB5	PORTB4	PORTB3	PORTB2	PORTB1	PORTB0	PORTB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

13.4.3 DDRB – The Port B Data Direction Register

Bit	7	6	5	4	3	2	1	0	
0x04 (0x24)	DDRB7	DDRB6	DDRB5	DDRB4	DDRB3	DDRB2	DDRB1	DDRB0	DDRB
Read/Write	R/W	R/W	R/W	R/W	R/W	R/W	R/W	R/W	
Initial Value	0	0	0	0	0	0	0	0	

```
void setup() {
  // initialize digital pin LED_BUILTIN as an output.
  DDRB = 32; //00100000
}

void loop() {
  PORTB = 32;
  delay(2000);
  PORTB = 0;
  delay(1000);
}
```

Delay

```
void loop() {  
  PORTB = 32;  
  
  for (long i = 0; i < 1000000; i++) {}  
  
  PORTB = 0;  
  
  for (long i = 0; i < 1000000; i++) {}  
}
```

This does not work :(

Arduino's compiler is too smart.

It sees that this loop does nothing, so it ignores it to optimize speed.

We need to trick the compiler to think we're doing something that it can't just ignore