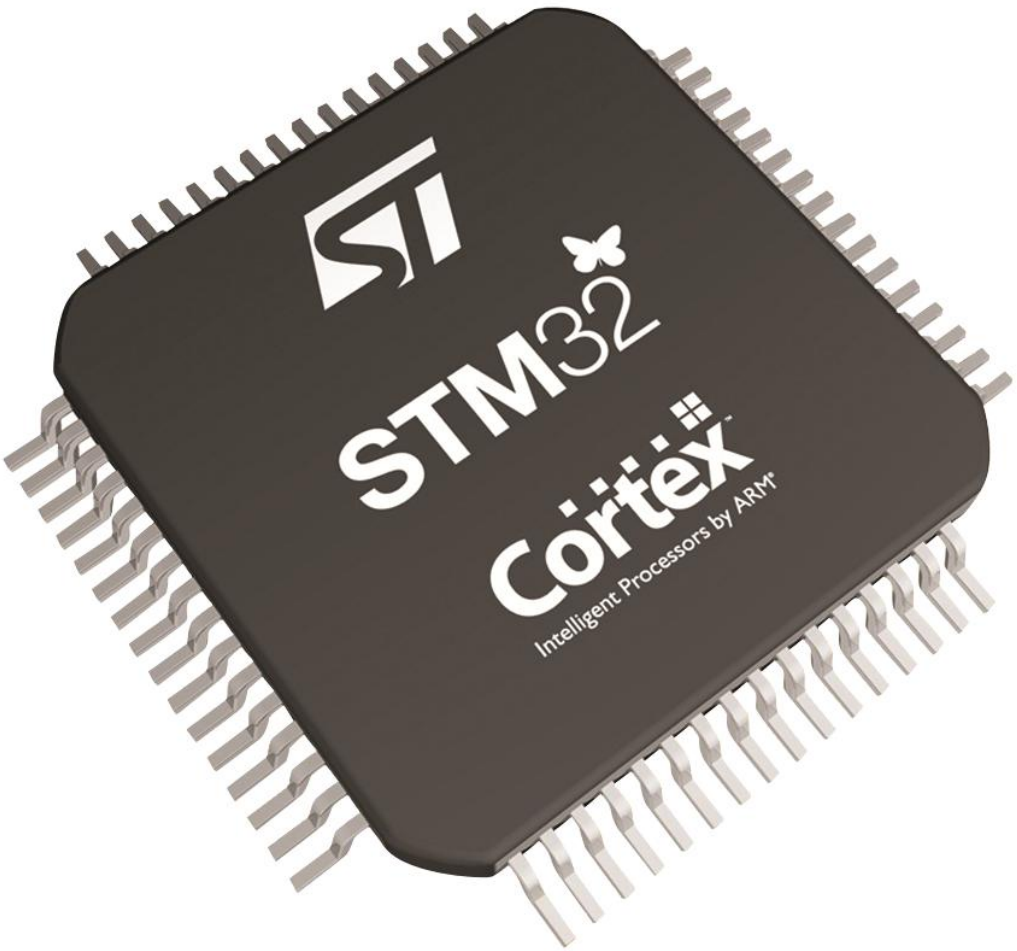
Embedded Systems



Contents

1 Notes

```
1  int var(0); // Declare variable as int and set its value to 0. This is ...
       the preferred method for assigning a value to variables in C++.
2
3  // Increment and decrementing a variable
4  // Post
5  a++
6  a--
7
8  // Pre
9  ++a
10 --a
11
12 // for loop example
13 for(int i = 0; i < 42; i++)
14 {
15     std::cout << i << std::endl
16 }
17
18 // Enumerator
19 enum FuelStatus{critical, low, normal, topped}
20
21 FuelStatus f = low;
22
23 switch(f)
24 {
25     case critical:
26         // Handle critical
27         break;
28     case low:
29         // Handle low
30         break;
31     // Etc...
32     default:
33         // Handle everything other than the alternatives in FuelStatus
34 }
35
36 // Functions
37 // type = return type, ex int, float etc...
38 type name(parameter1, parameter2, ...) {statements}
39
40 // A function that don't return anything
41 void name(parameter1, parameter2, ...) {statements}
```

Code 1.1: Lecture 2-3

```
// Pekere etc:
void increment(&valAddr)
{
    (*valAddr)++;
}


int value = 5;
int* valueAddr = &value;

increment(&value);

// Referanser
void increment(int& v)
{
    v++;
}


int value = 5;
int* valueAddr = &value;

increment(value);
```

Code 1.2: Lecture 5-6

```
// OOP!

// Klasser i C++
// Kan ha:
// - Medlemsvariable
// - Medlemsfunksjoner ("Metoder")

// Klasser i enkleste form

class WayPoint
{
    public:
        WpType type:
        double x;
        double y;
};

// Nesten det samme som struct

struct WayPoint
{
    // Default is public here
    WpType name;
    double x;
    double y;
};
```

Code 1.3: Lecture 7

https://en.cppreference.com/w/

1.1 Pointers and refrences

Notation	Description
int* pointer	Creates a pointer named pointer
int *ptrA, *ptrB	Creates two pointers, prtA and ptrB
pointer = &cat	Stores the memory address of variable cat in pointer (the place in memory where the variable cat is stored
value = *pointer	Dereferencing the pointer so that we get the actual value stored in memory

Table 1.1: Pointers and references

1.2 Classes, header & cpp files

Create new class via File $\longrightarrow$ New file $\longrightarrow$ C++ class in Qt (generates both header file and cpp file with default template)