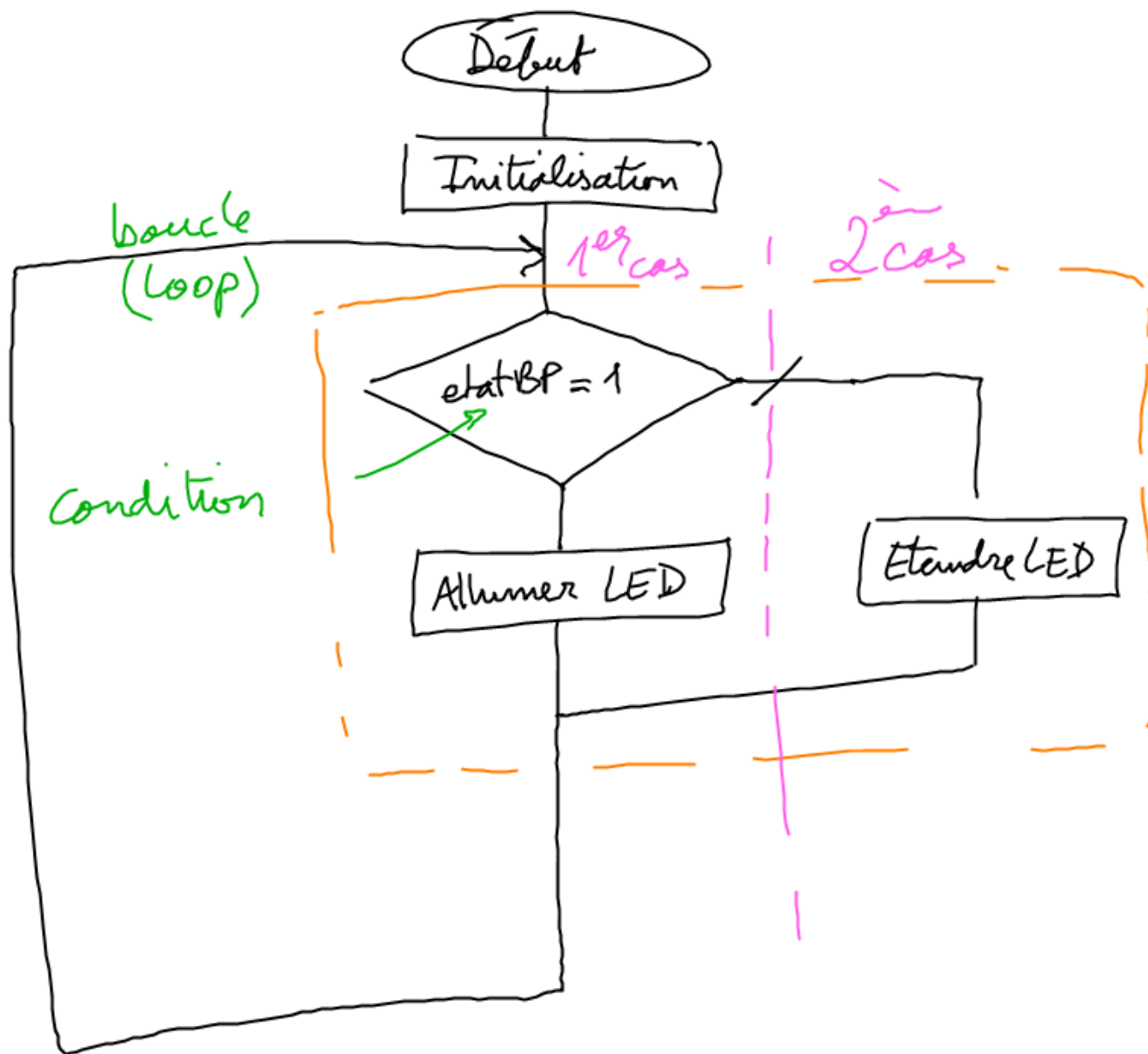




Structure

Si -- alors -- sinon

Temps



// l'en-tête déclarative permet de définir, si nécessaire, des variables

```
int broche_LED = 3;
```

```
int broche_BP = 2;
```

```
int etat_BP;
```

variable

// la fonction setup() est exécutée en premier et une seule fois, au démarrage du programme

```
void setup() {
```

```
    pinMode(broche_LED, OUTPUT); // initialisation de la broche 3 en sortie
```

```
    pinMode(broche_BP, INPUT); // initialisation de la broche 2 en entrée
```

```
}
```

boucle // la fonction loop() s'exécute sans fin en boucle aussi longtemps que la carte Arduino est sous tension

```
void loop() {
```

```
    etat_BP = digitalRead(broche_BP); // enregistre l'état du bouton poussoir dans la variable etat_BP
```

```
    if(etat_BP == 1) {
```

```
        digitalWrite(broche_LED, 1); // mettre la broche 3 au niveau logique haut = mettre à 5V la broche
```

```
    }
```

```
    else
```

```
    {
```

```
        digitalWrite(broche_LED, 0); // mettre la broche 3 au niveau logique bas = mettre à 0V la broche
```

```
    }
```

```
}
```

```

int utemps = 150; // en ms
int tcourt = utemps;
int tlong = utemps * 3;
int tentre = utemps;
int tlettre = utemps * 3;
int tmot = utemps * 7;

```

```

digitalWrite(broche_LED, 1);
delay(tcourt);
digitalWrite(broche_LED, 0);
delay(tentre);
digitalWrite(broche_LED, 1);
delay(tcourt);
digitalWrite(broche_LED, 0);
delay(tentre);
digitalWrite(broche_LED, 1);
delay(tcourt);
digitalWrite(broche_LED, 0);
delay(tlettre);

digitalWrite(broche_LED, 1);
delay(tlong);
digitalWrite(broche_LED, 0);

```

```

int broche_LED = 3;
int broche_BP = 2;
int etat_BP; variable

// la fonction setup() est exécutée
void setup() {

    pinMode(broche_LED, OUTPUT); //
    pinMode(broche_BP, INPUT); //

}

// la fonction loop() s'exécute sans
void loop() {

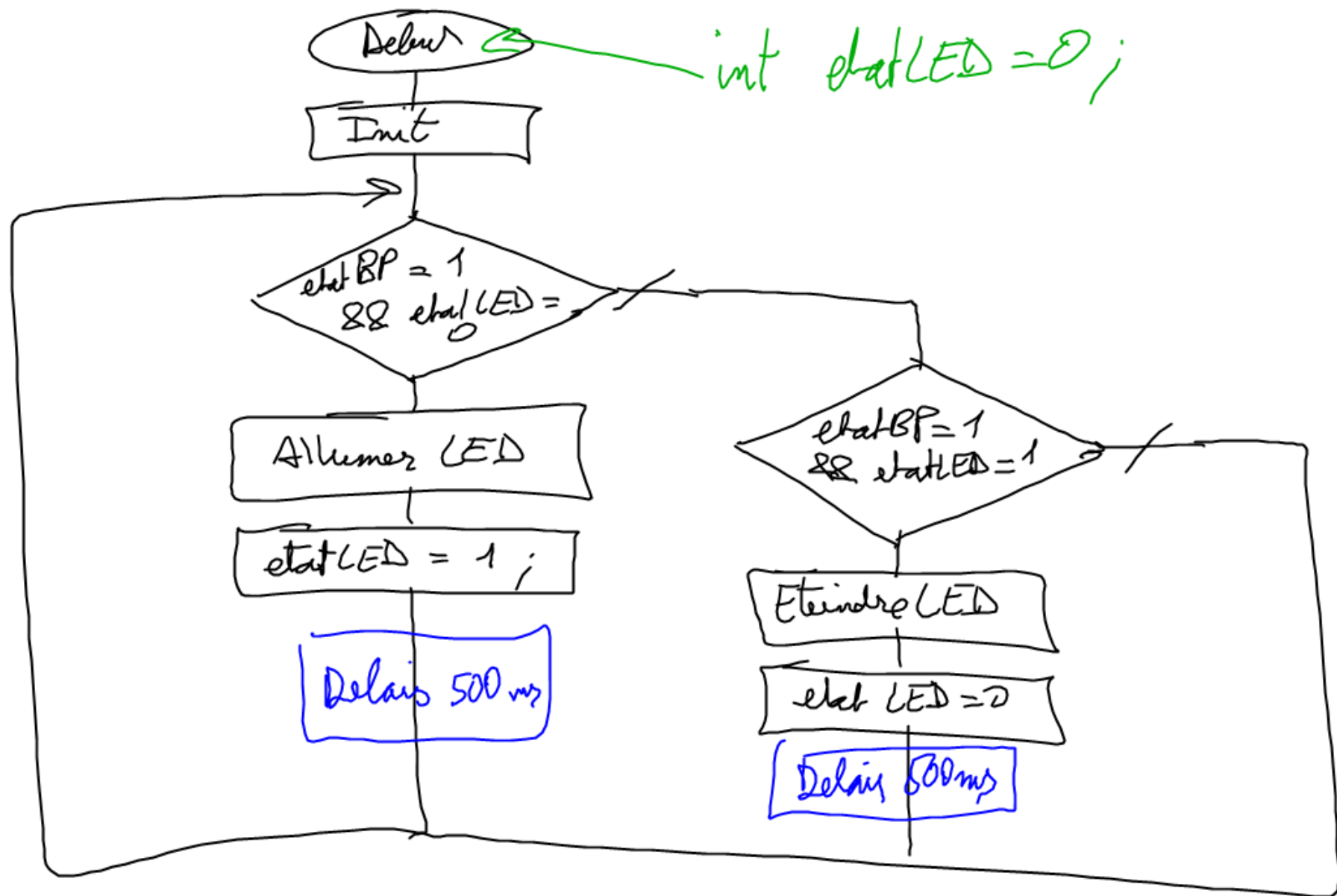
    etat_BP = digitalRead(broche_BP);

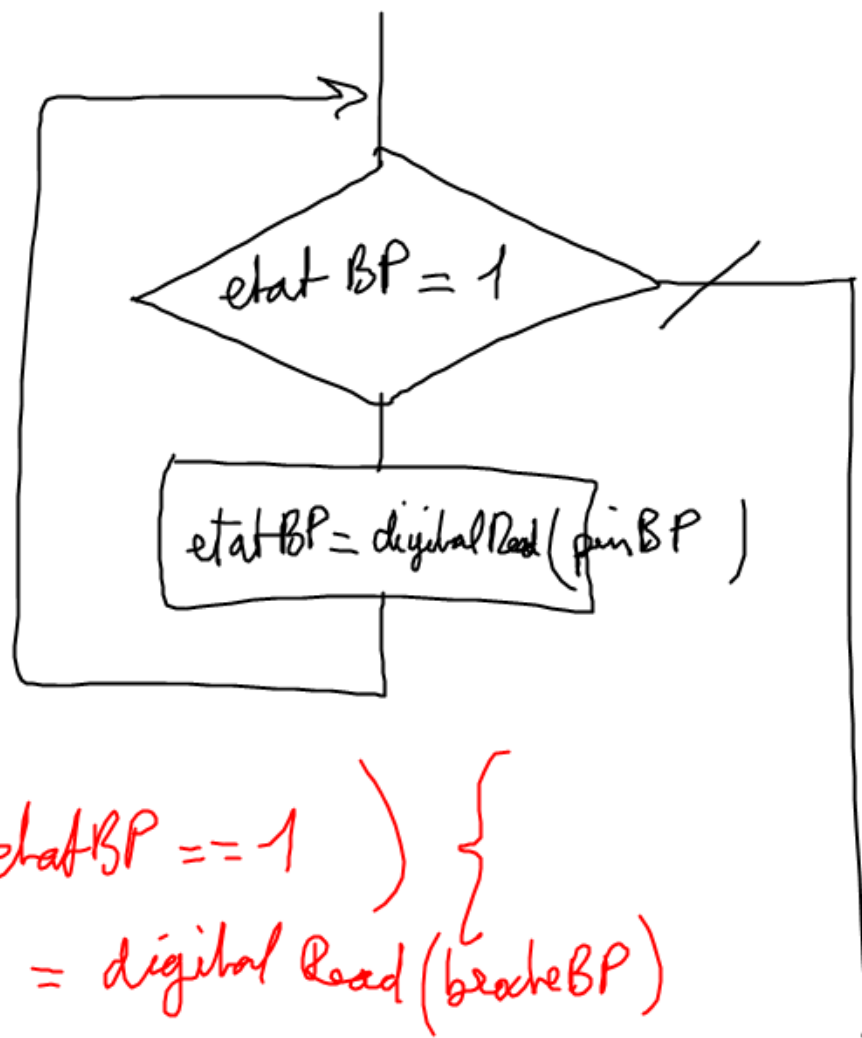
    if(etat_BP == 1) { condition
        digitalWrite(broche_LED, 1);
    }
    else
    {
        digitalWrite(broche_LED, 0);
    }
}

```

boucle

et : 22 esperluette
ou : 11 grep





```

while (etatBP == 1) {
  etatBP = digitalRead(pinBP)
}
  
```

tant que
faire

while

! bloquant



7/7

BP_mem\$

```
etat_BP = digitalRead(broche_BP);

if (etat_BP == 1 && etat_LED == 0) {
    digitalWrite(broche_LED, 1);
    etat_LED = 1;
    while(etat_BP == 1) {
        etat_BP = digitalRead(broche_BP);
    }
}
else if (etat_BP == 1 && etat_LED == 1)
{
    digitalWrite(broche_LED, 0);
    etat_LED = 0;
    while(etat_BP == 1) {
        etat_BP = digitalRead(broche_BP);
    }
}
}
```

Compilation terminée.

Le croquis utilise 946 octets (2%) de l'espace de stockage de programmes. Le maximum est de 32256 octets.
Les variables globales utilisent 11 octets (0%) de mémoire dynamique, ce qui laisse 2037 octets pour les variables

