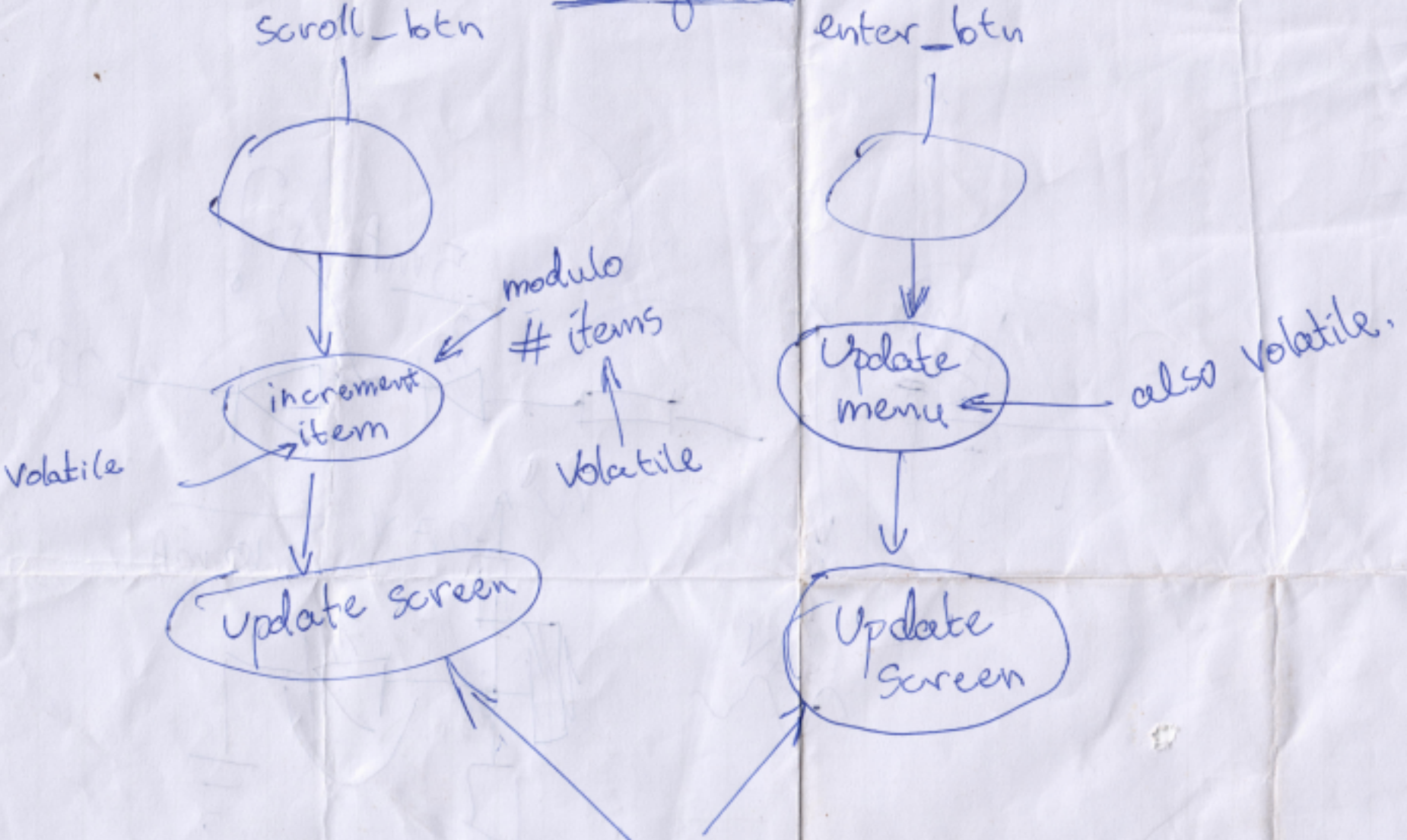
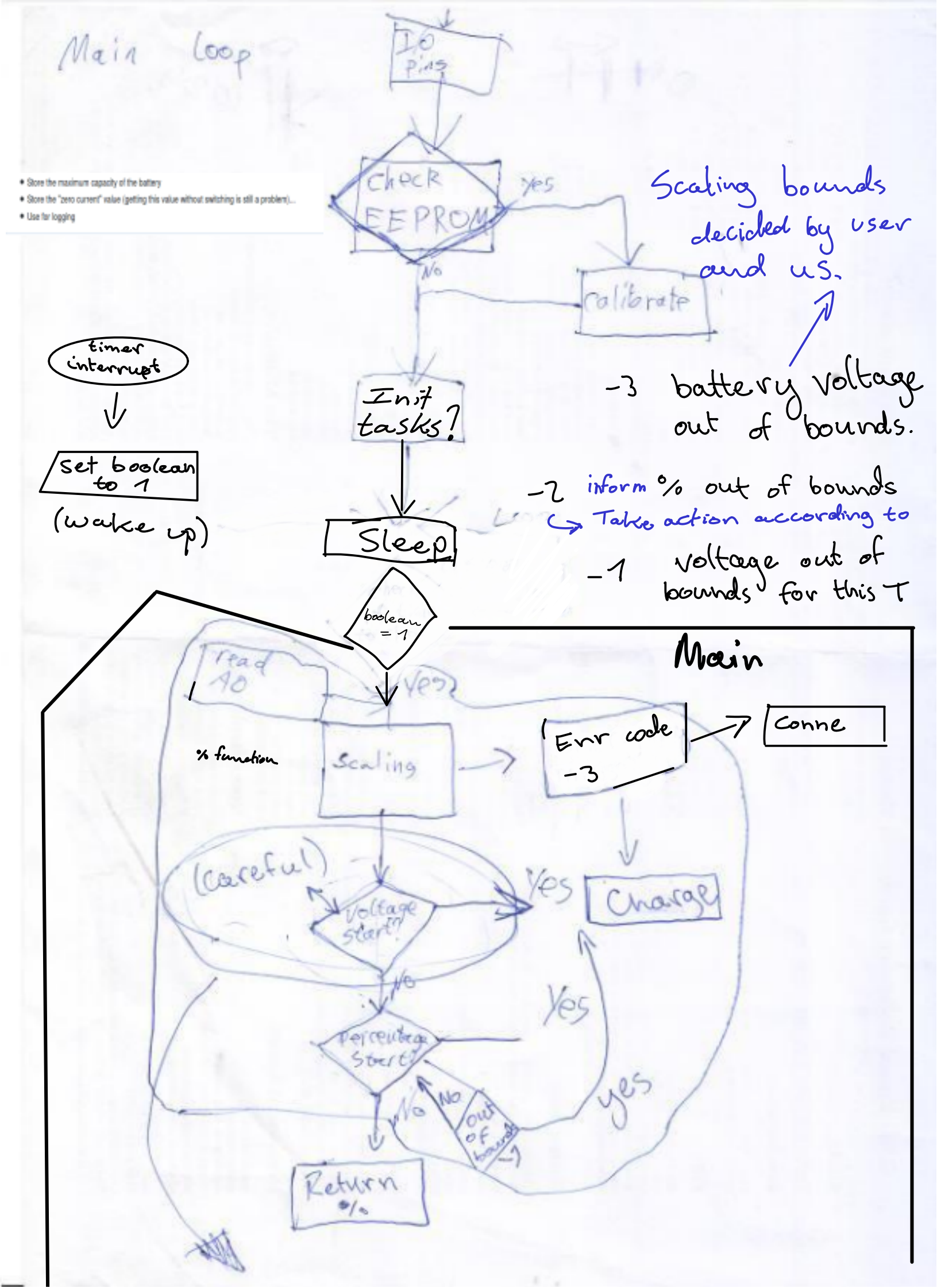
Navigation



Critical section: increment item.

Solution: Prioritize btn
or disable interrupts (if no prioritization).

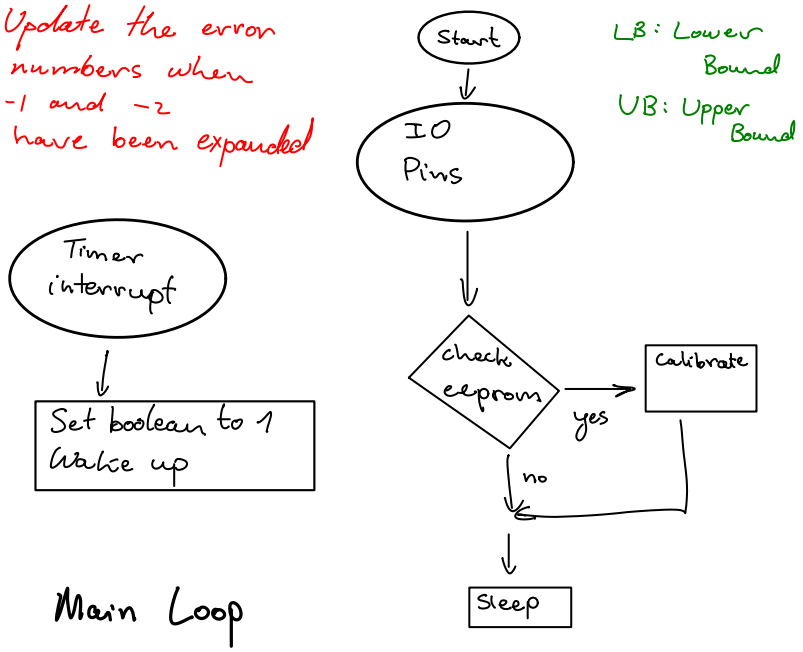


- Store the maximum capacity of the battery
- Store the "zero current" value (getting this value without switching is still a problem)...
- Use for logging

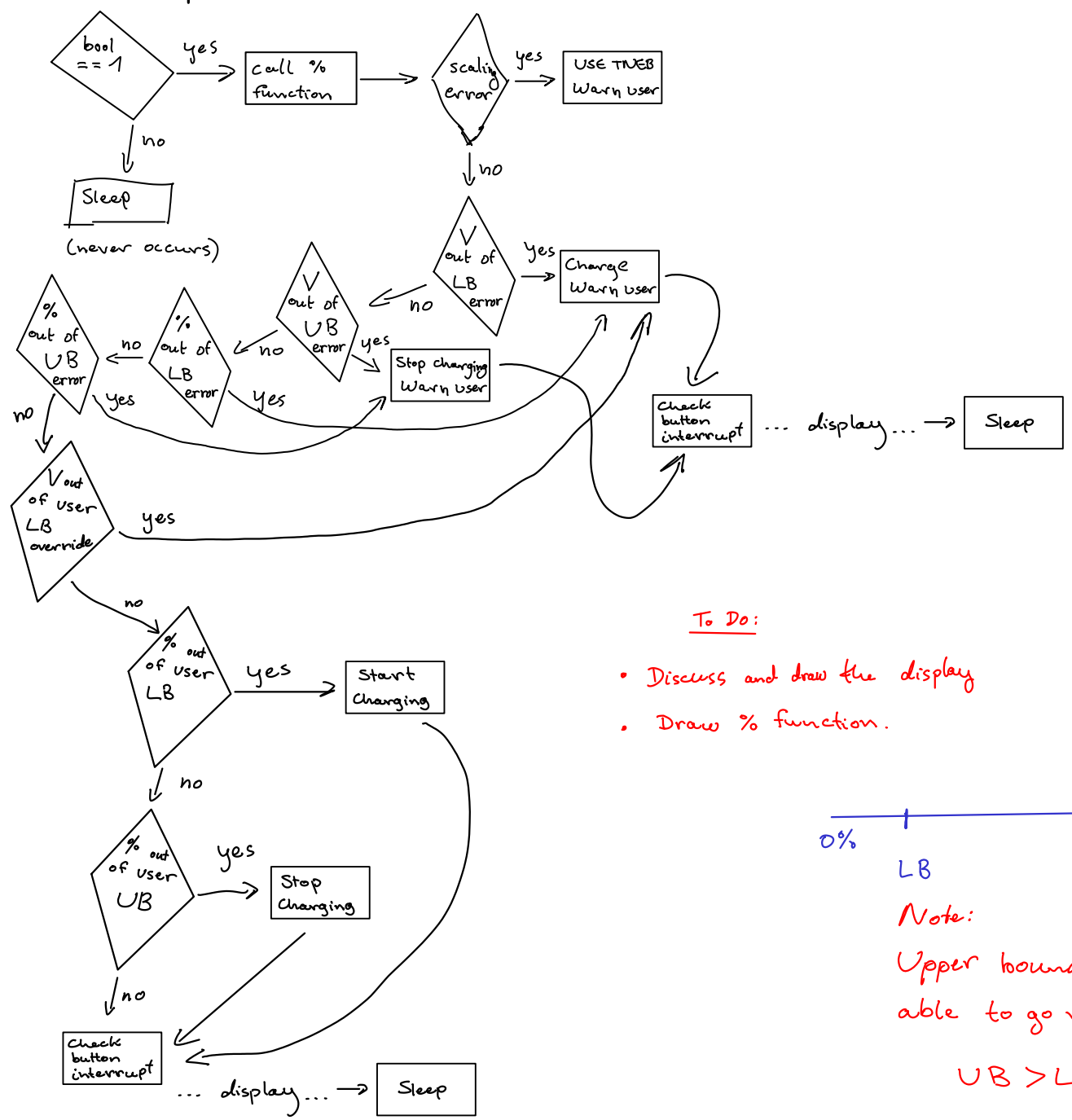
Update the error numbers when -1 and -2 have been expanded

LB: Lower Bound

UB: Upper Bound



Main Loop



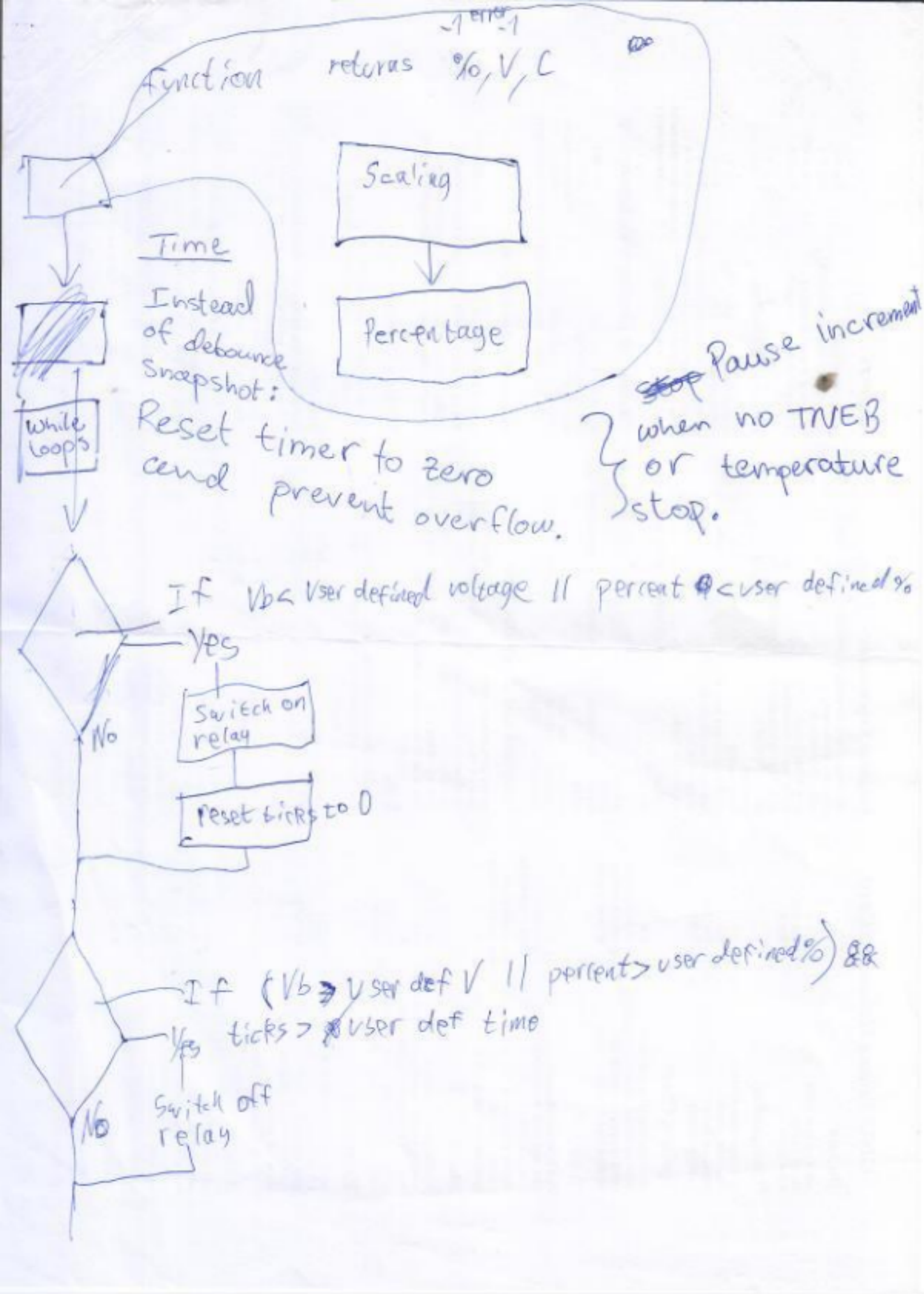
To Do:

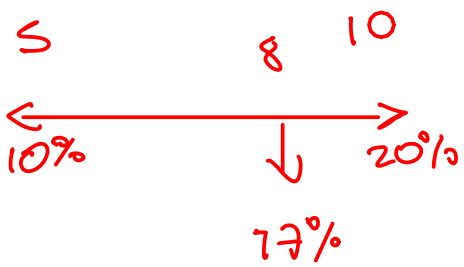
- Discuss and draw the display
- Draw % function.

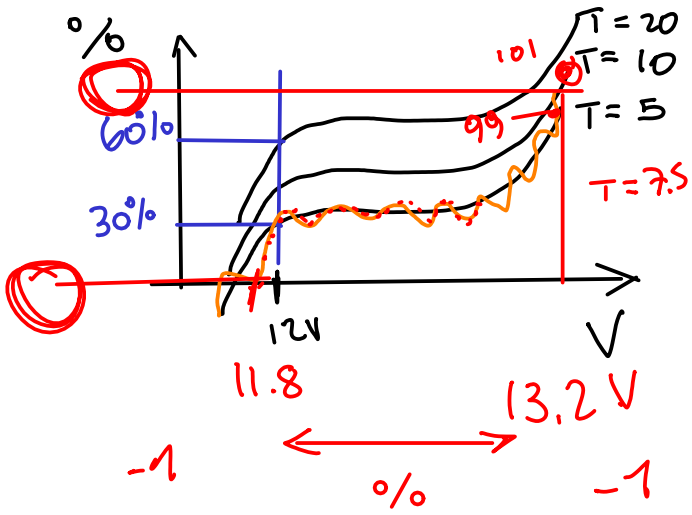
Note:

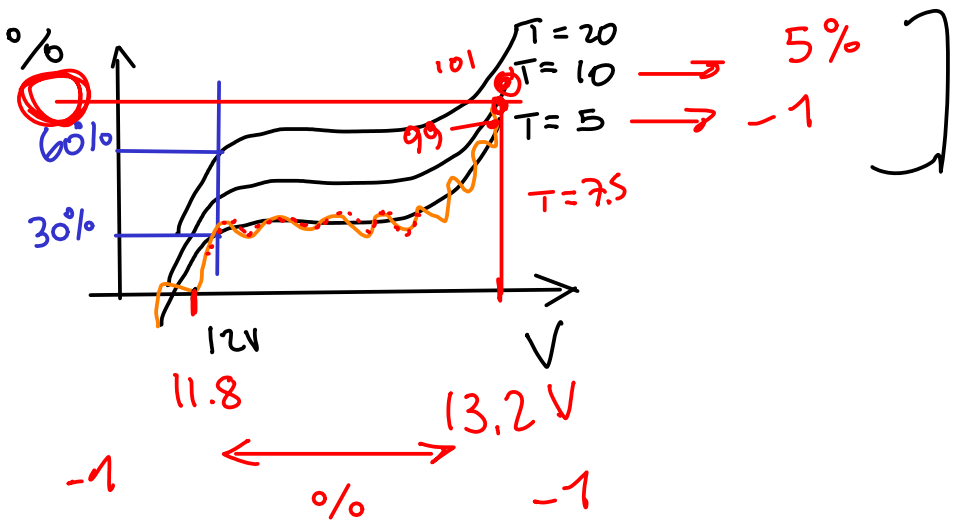
Upper bound should be able to go very far down.

UB > LB + buffer


-1 error -1
function returns %, V, C
Scaling
Percentage
Time
Instead of debounce
Snapshot:
Reset timer to zero
and prevent overflow.
while loop
Pause increment when no TNEB or temperature stop.
If Vb < user defined voltage || percent < user defined %
Yes
No
Switch on relay
Reset ticks to 0
If (Vb > user def V || percent > user defined %) &&
ticks > user def time
Yes
No
Switch off relay

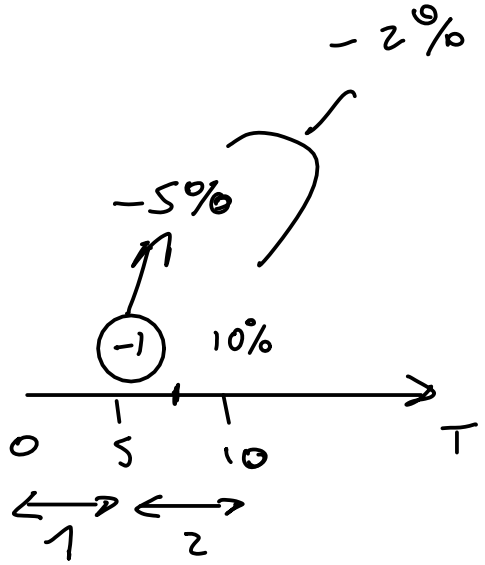







```
if (T <= 5) {
  // 5 Formula: (x - 17.481) * (x - 14.408) * (x - 14.202) * (x - 12.168) * (x - 17.452) * (x - 14.783) + 78.465
  // Serial.print("< 5 ");

  if (12.64 < U && U < 15.77) {
    float result = (U - 17.481) * (U - 14.408) * (U - 14.202) * (U - 12.168) * (U - 17.452) * (U - 14.783) + 78.465;

    if (0 <= result && result <= 100) {
      return result;
    } else {
      return -2;
    }
  } else {
    return -1;
  }
```

} Errors




```
} else if (5 < T && T <= 10) {
  // 10 Formula: (x - 11.992) * (x - 14.325) * (x - 17.428) * (x - 14.420) * (x - 14.353) * (x - 17.227) + 88.222)
  // return ((T - 5) * (10 formula)  + (10 - T) * (5 formula) / 5
  // Serial.print("5 < T && T < 10 ");

  if (12.49 < U && U < 15.77) {

    float result = ((T - 5) * ((U - 11.992) * (U - 14.325) * (U - 17.428) * (U - 14.420) * (U - 14.353) * (U - 17.227) + 88.222)
          + (10 - T) * ((U - 17.481) * (U - 14.408) * (U - 14.202) * (U - 12.168) * (U - 17.452) * (U - 14.783) + 78.465)) / 5;

    if (0 <= result && result <= 100) {
      return result;
    } else {
      return -2;
    }
  } else {
    return -1;
  }
```

0

T_{10} →

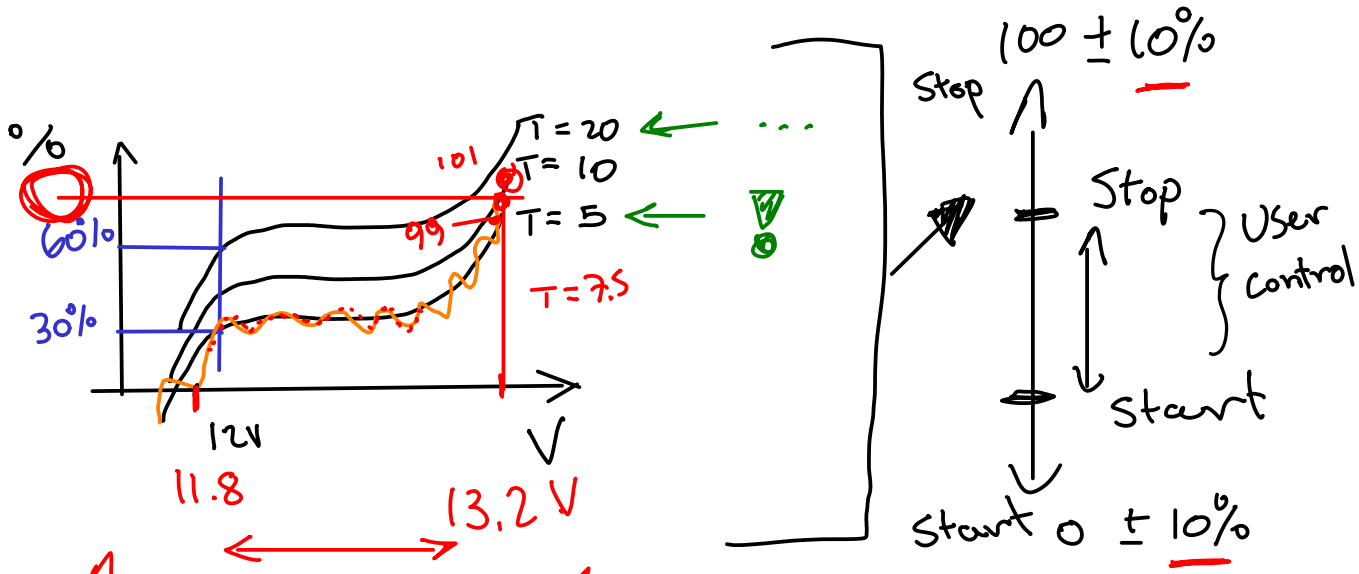
T_5 →

Split into lower and upper bounds

Check the error codes after calculating the average.

- 1.1
- 1.2
- 2.1
- 2.2
- 3

} 5 errors.



-1 ↑ Charge!

% ↓ Display error message with V and I?

-1 ↑ Stop charging!

Different errors!

```
if (T >= 200) {
  // Rest

  if (12.69 > U && U > 11.77) {
    float result = (U - 11.4656) * (U - 14.9852) * (U + 6.5100) * (U - 11.1390) * (U - 14.1120) * (U - 12.0202) + 20.7968;

    if (0 <= result && result <= 100) {
      return result;
    } else {
      return -2;
    }
  } else {
    return -1;
  }


} else if (T < 3) {
  // 3 formula: (x - 10.1354) * (x - 10.1681) * (x - 10.1423) * (x - 6.4437) * (x - 13.7005) * (x - 13.5256) + 27.9198
  // return 3 formula;

  if (11.71> U && U > 9.35) {
    float result = (U - 10.1354) * (U - 10.1681) * (U - 10.1423) * (U - 6.4437) * (U - 13.7005) * (U - 13.5256) + 27.9198;

    if (0 <= result && result <= 100) {
      return result;
    } else {
      return -2;
    }
  } else {
    return -1;
  }
}
```

→ Slow discharge

→ Fast discharge

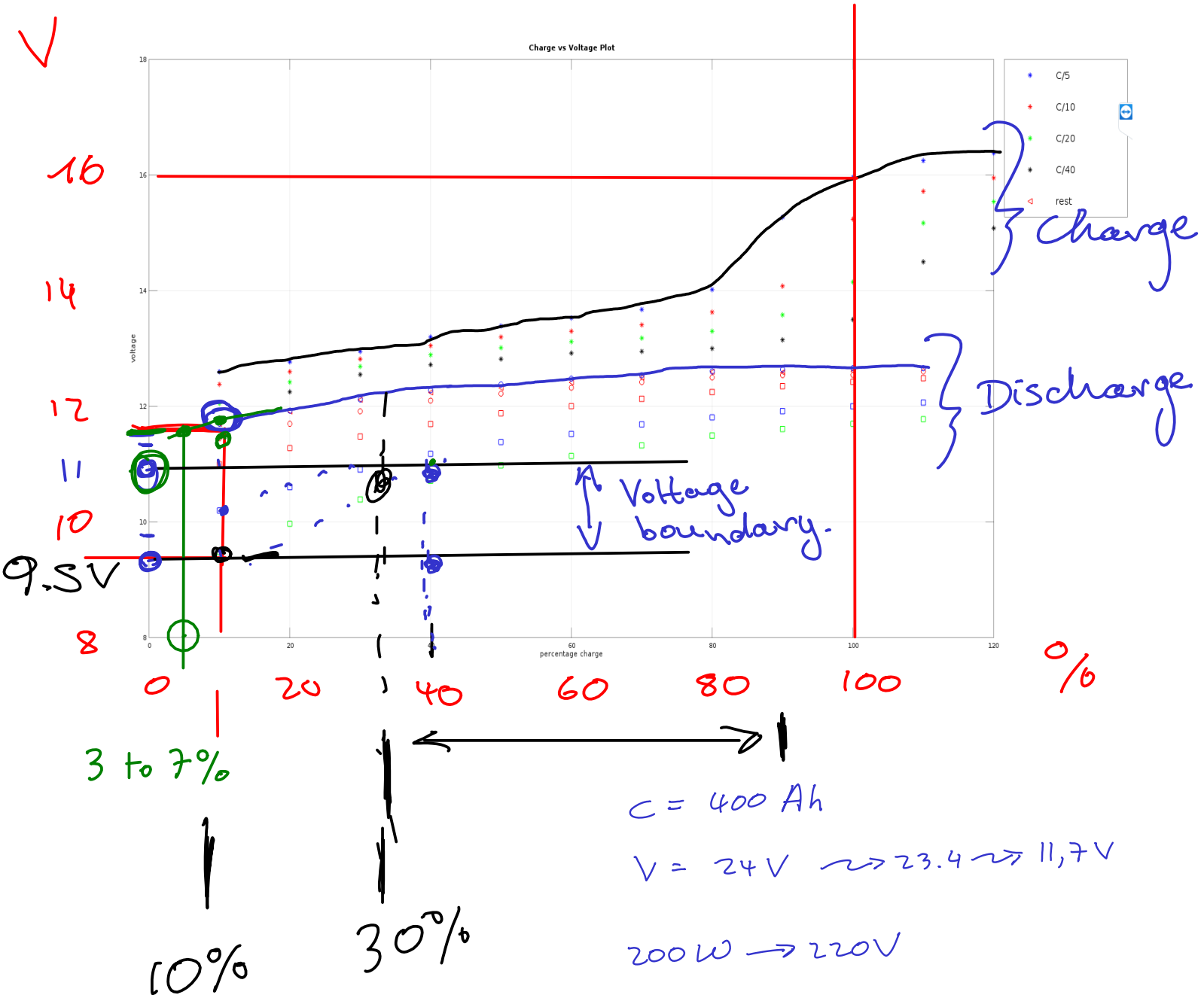
UPS off at 11 V?

```
// Can't be right: scaling error
if (V < 8.5 || V > 15.5) {
  percent = -3;
} else {
  percent = percentage(V, I, C);
}
```

Minimise ← [-3 error overrides?? ; - UPS also overrides]

↓

Unavoidable?



$C = 400\ Ah$

$V = 24\ V \rightsquigarrow 23.4 \rightsquigarrow 11{,}7\ V$

$200\ W \rightarrow 220\ V$

$\frac{200}{24} = 8.3\ A$

$400\ Ah \rightarrow 1\ h \quad 400\ A$

$1\ h \cdot \frac{400}{8{,}3} = 48\ h$ till empty

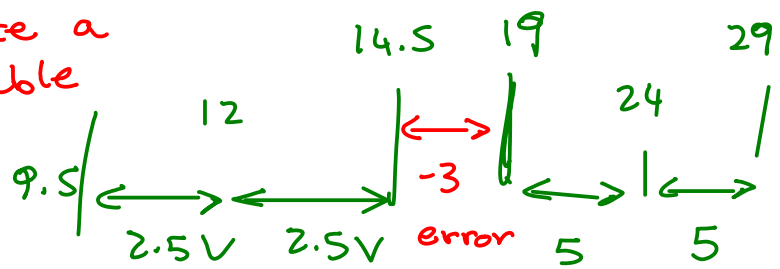
$\Rightarrow C/40$

```
// Voltage measurement and scaling
float measuredVoltage = batteryVoltage();
int divisor = (int) (measuredVoltage / 9.5);
if (divisor < 1) {
  divisor = 1;
}
// Scaled voltage:
float V = measuredVoltage / divisor;

// Current measurement - no scaling
float I = currentInAmps();

// Can't be right: scaling error
if (V < 8.5 || V > 15.5) {
  percent = -3;
} else {
  percent = percentage(V, I, C);
}
```

$\frac{25}{9.5} = 2.5$

$\frac{25}{2} = 12.5$

Create a variable



Not using -3 error to override charging just use it for scaling.

Reason → Also overrides high side / Upper bound.

↓

Also move specific error message.