

Félix-Antoine Guimont  
Marc-Étienne Gendron-Fontaine  
Groupe : 2317

## **CANVision**

Manuel technique présenté à :  
Benoît Beaulieu  
Simon-Pierre Gagnon  
Louis-Philippe Gauthier  
Université de Sherbrooke (FUSE)

Département du génie électrique

Pour le cours  
Projet de fin d'études

## Table des matières

<b>1.</b>	<b>Description générale .....</b>	<b>3</b>
<b>1.1</b>	<b>À propos .....</b>	<b>3</b>
<b>1.2</b>	<b>Explication générale du projet .....</b>	<b>4</b>
<b>2</b>	<b>Fonctionnement .....</b>	<b>5</b>
<b>3</b>	<b>Procédure d'installation et d'opération .....</b>	<b>7</b>
<b>3.1</b>	<b>Installation du système .....</b>	<b>7</b>
<b>3.2</b>	<b>Comment opérer l'écran .....</b>	<b>9</b>
<b>4</b>	<b>Contenu matériel .....</b>	<b>10</b>
<b>4.1</b>	<b>Régulateur de tension .....</b>	<b>10</b>
<b>4.2</b>	<b>Connecteur CAN Bus et Transmetteur.....</b>	<b>10</b>
<b>4.3</b>	<b>Connecteur de l'écran .....</b>	<b>11</b>
<b>5</b>	<b>Contenu logiciel .....</b>	<b>12</b>
<b>5.1</b>	<b>Logiciel Utilisé .....</b>	<b>12</b>
<b>5.2</b>	<b>Librairie utilisé .....</b>	<b>12</b>
<b>5.3</b>	<b>Compréhension du code principale.....</b>	<b>13</b>
<b>6</b>	<b>Procédure de développement .....</b>	<b>14</b>
<b>6.1</b>	<b>Téléverser dans l'écran .....</b>	<b>14</b>
<b>6.2</b>	<b>Installation des logiciels et des librairies .....</b>	<b>14</b>
<b>6.3</b>	<b>Modifier l'affichage du projet.....</b>	<b>17</b>
<b>6.4</b>	<b>Modifier le code CAN .....</b>	<b>18</b>
<b>7</b>	<b>Listes de matériel et coûts .....</b>	<b>20</b>
<b>8</b>	<b>Modifications et améliorations .....</b>	<b>23</b>
<b>9</b>	<b>Annexe .....</b>	<b>24</b>
<b>9.1</b>	<b>Annexe 1 : Schémas électrique .....</b>	<b>24</b>

## 1. Description générale

### 1.1 À propos

Le projet consiste à afficher des informations dans le but de diagnostiquer une voiture de Formule SAE durant la compétition aux États-Unis. Mais c'est quoi une Formule SAE ? Ce sont des épreuves de voiture de course monoplace. Cette compétition a pour objectif la formation de futurs professionnels à l'ingénierie de la mobilité. Il y a quatre catégories de véhicule pour cette compétition : véhicule thermique, hybride, électrique et autonome.



Figure 1: Voiture de Formule SAE

Ce projet a démarré à l'automne 2023 et nous sommes la première équipe à faire ce projet. Le but est de créer une interface de diagnostic pour les mécaniciens de la monoplace. Les clients sont des étudiants de l'Université de Sherbrooke en Génie Électrique. Notre projet est d'installer un écran dans la partie arrière de la voiture pour aider les mécaniciens durant la compétition. Cet écran est connecté au réseau CAN de la monoplace.

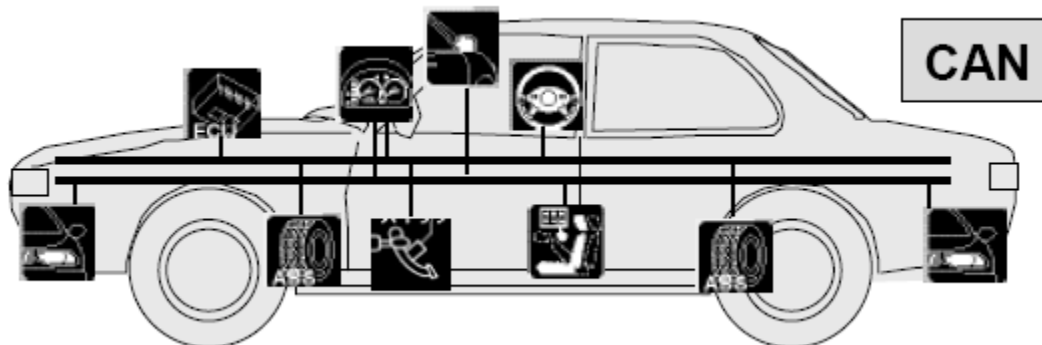


Figure 2: Illustration du Réseau CAN

## 1.2 Explication générale du projet

Le but du projet est d'afficher les données des capteurs pour pouvoir faire des diagnostics de la monoplace. Présentement, nous utilisons un affichage de sept pouces de la compagnie 4D system. Aussi, nous avons réalisé un PCB qui réceptionne le réseau CAN. Ce PCB est alimenté avec du 12V de la Drive du véhicule. Les données reçues sont mises à jour en temps réel (0,2 seconde) sur l'écran. Sur cette dernière, nous avons trois pages de diagnostics : HV (accumulateur), Temp (Température), LV (Batterie). Ces trois pages contiennent environ 3 ou 4 widgets. Par exemple, la page HV contient 4 widgets qui vérifient sa santé, sa charge, sa température et sa tension.

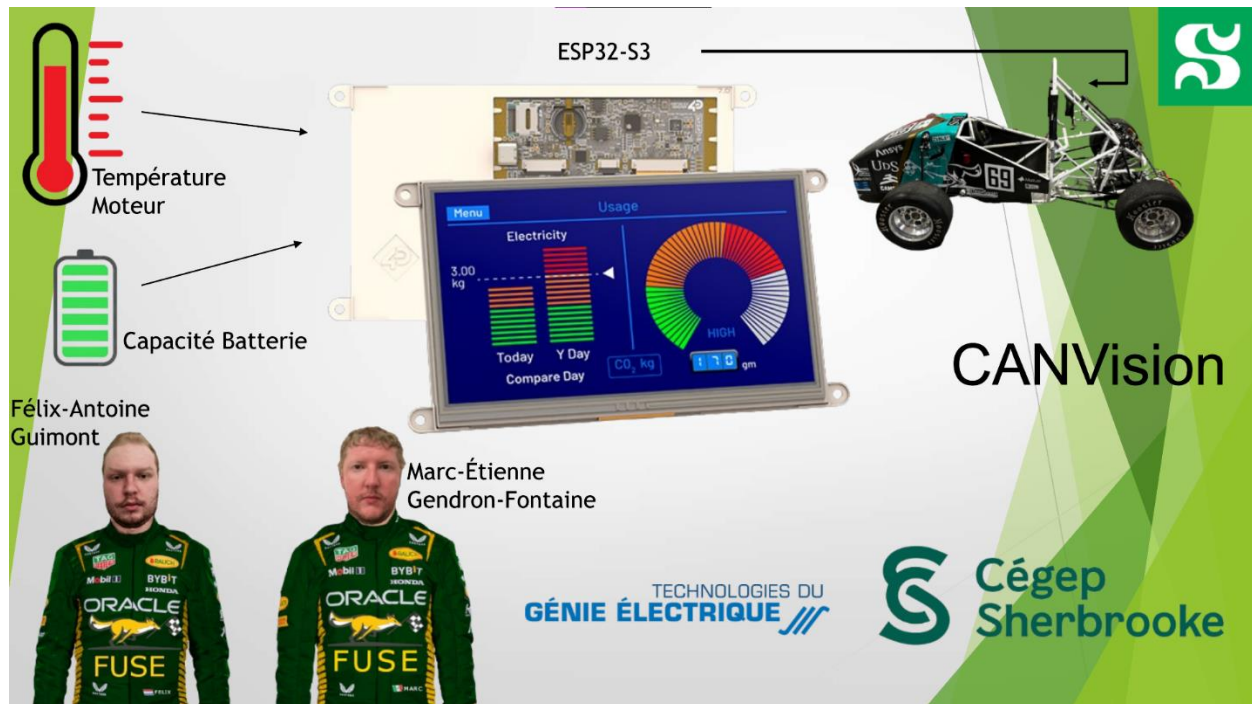


Figure 3: Illustration du projet

## 2 Fonctionnement

Sur le schéma ci-dessous, on remarque que le projet est constitué de deux sections principales. Le PCB et l'écran.

- Le PCB s'occupe de s'interfacer au CAN bus pour convertir les signaux électriques vers les signaux CAN RX et CAN TX qui seront lus par l'ESP32 de l'affichage.
- L'écran contient un ESP32-S3R8. Il reçoit les données en série CAN.

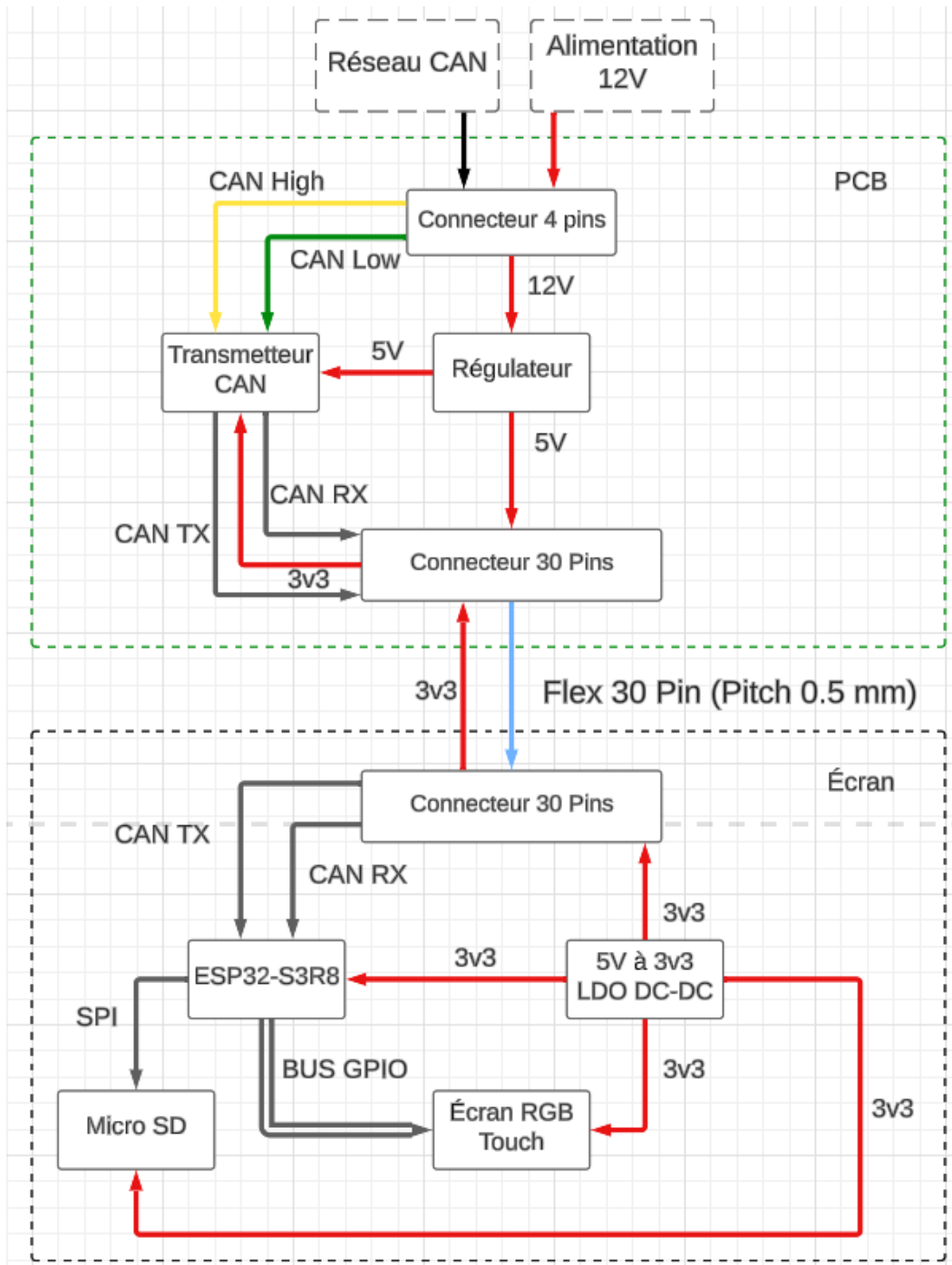


Figure 4: Schémas conceptuel

Le réseau CAN en différentiel et l'alimentation de 12V sont fournis dans le connecteur 4 pin vers le PCB.

Le régulateur (LM2675M-5.0), qui est une alimentation à découpage, prend l'alimentation de 12V provenant de la batterie et l'abaisse à 5V.

Le transmetteur (TJA1051T) reçoit les trames CAN. Ces trames contiennent des données de l'état de certaines composantes du véhicule telles que : l'accumulateur (HV), la batterie (LV) ainsi que des températures. Il reçoit une tension de 3V3 pour que le ESP32-S3R8 se communique bien ensemble. Le transmetteur envoie les trames reçues au ESP32-S3R8 via une communication série CAN.

Le flex de 30 pins fait la connexion entre le PCB et l'écran. Il alimente l'écran avec une tension de 5V.

Le ESP32-S3R8 reçoit une trame CAN. Il les traite les données pour les convertir en valeur entière pour que sur l'écran les données soient lisibles. Les trames sont reconnues en fonction des identifiants des capteurs. Ensuite, le ESP32-S3R8 met à jour les valeurs des widgets pour qu'ils soient visibles sur l'affichage.

### 3 Procédure d'installation et d'opération

#### 3.1 Installation du système

1. Connecter l'une des extrémités du flex sur le PCB et l'autre sur l'écran comme montré ci-dessous.



*Figure 5: connections interne*

2. Insérer l'écran sur le boîtier. L'écran devrait s'emboîter sur le boîtier comme montré ci-dessous.



*Figure 6: Vue extérieur*

3. Voici la disposition des broches du connecteur ATM13-4P : broche 1 GND, broche 2 CAN Low, broche 3 CAN High et broche 4 12V
- Broche 1 : GND
  - Broche 2 : CAN Low
  - Broche 3 : CAN High
  - Broche 4 : 12V

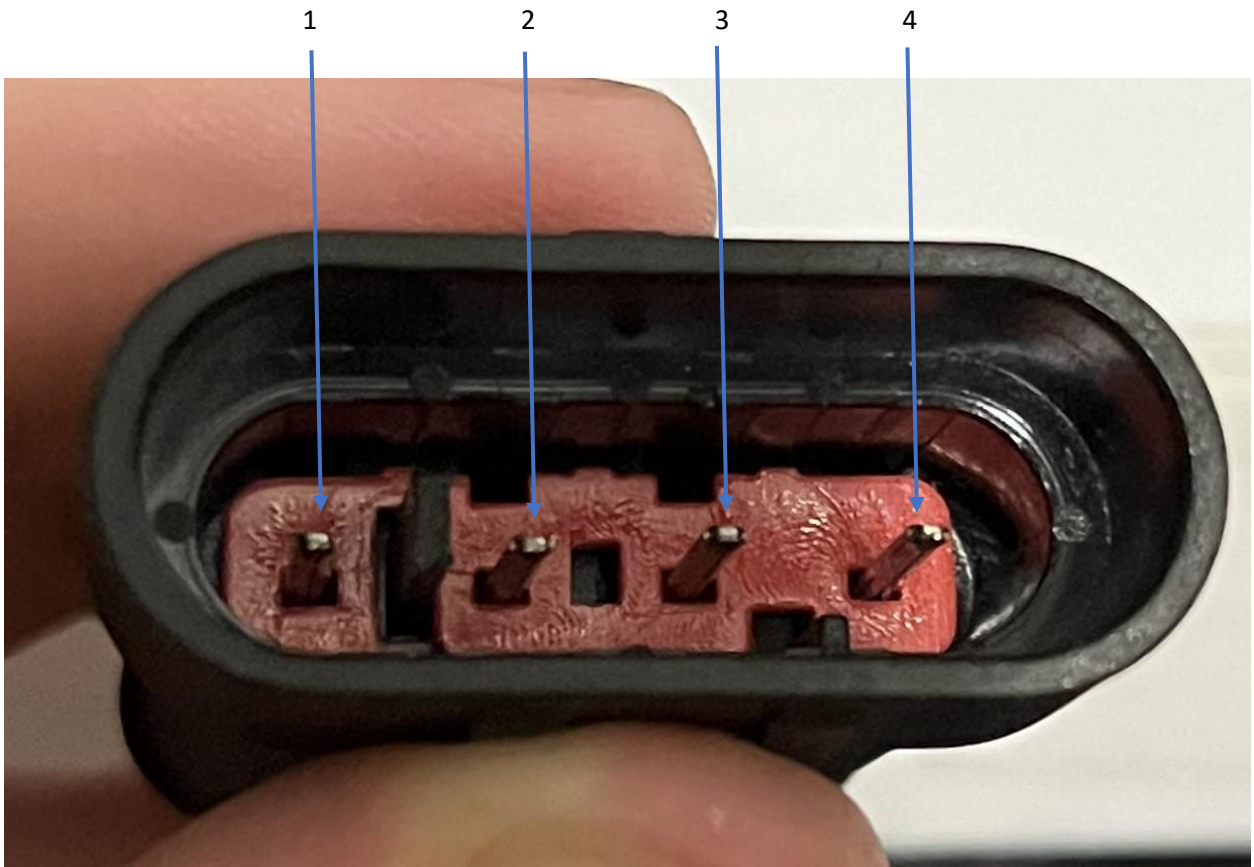
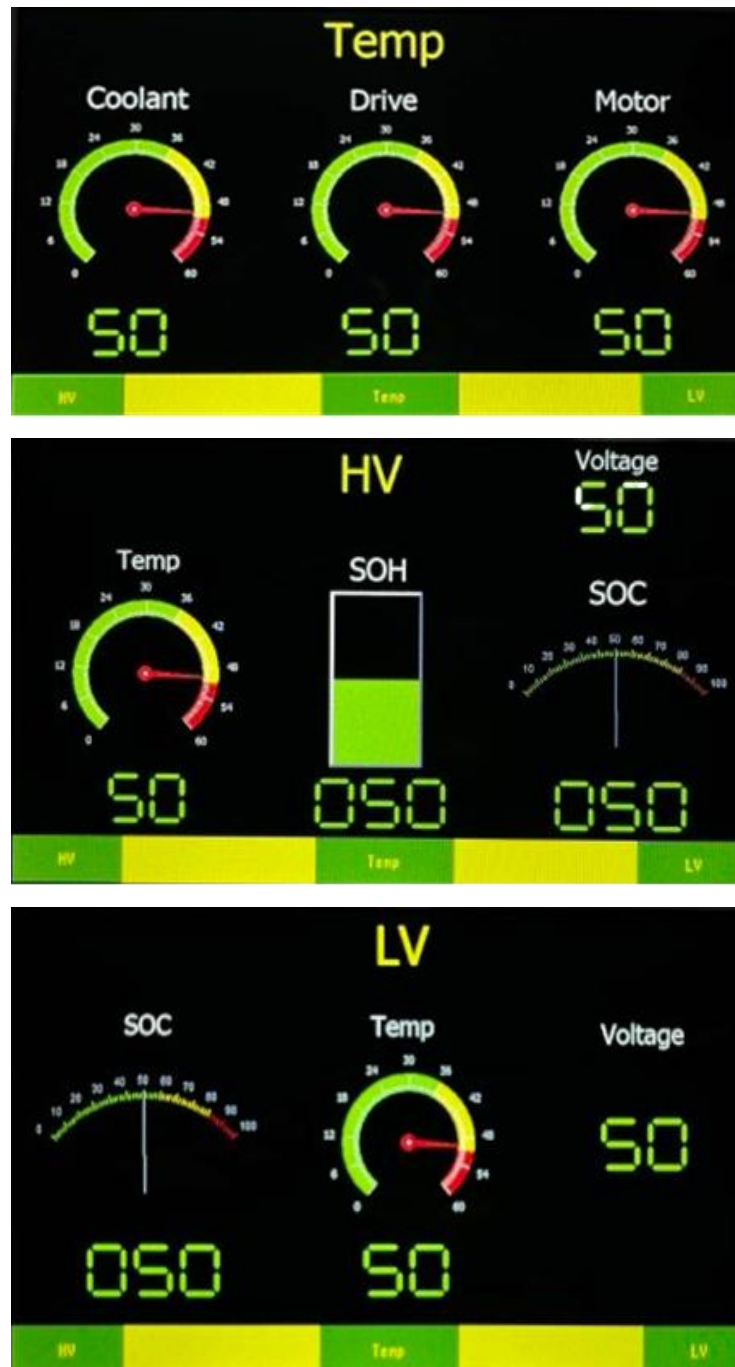


Figure 7: Connecteur



### 3.2 Comment opérer l'écran

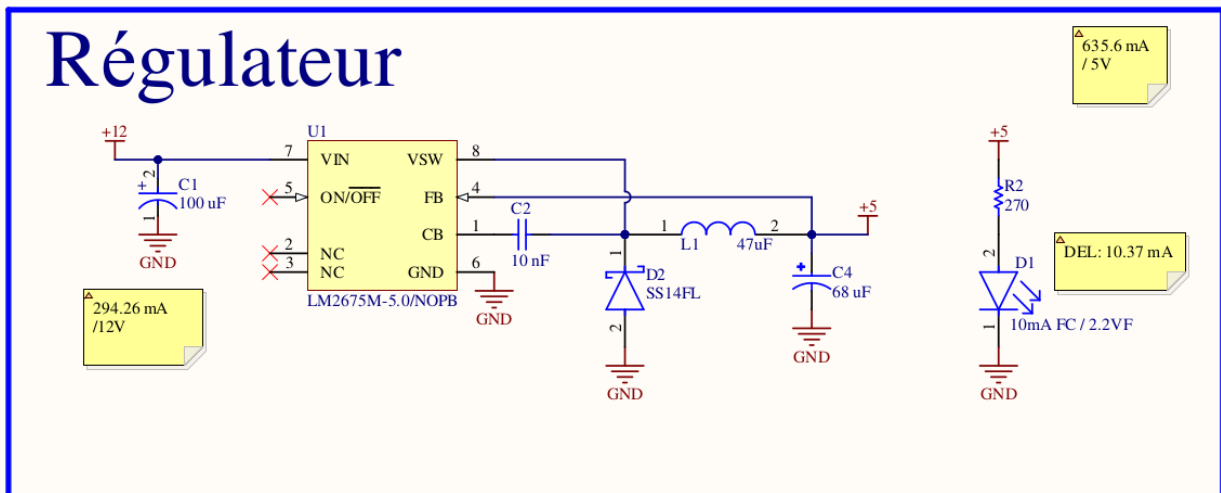
Une fois l'écran allumé, il affichera la page par défaut qui est « Temp ». Cependant, si l'écran ne détecte pas de réseau CAN, il y aura un message d'erreur qui affichera « NO Data CAN Recieved ». Pour changer de page, il faut simplement toucher le bouton correspondant. Les valeurs des widgets se mettent à jour de manière automatique.



## 4 Contenu matériel

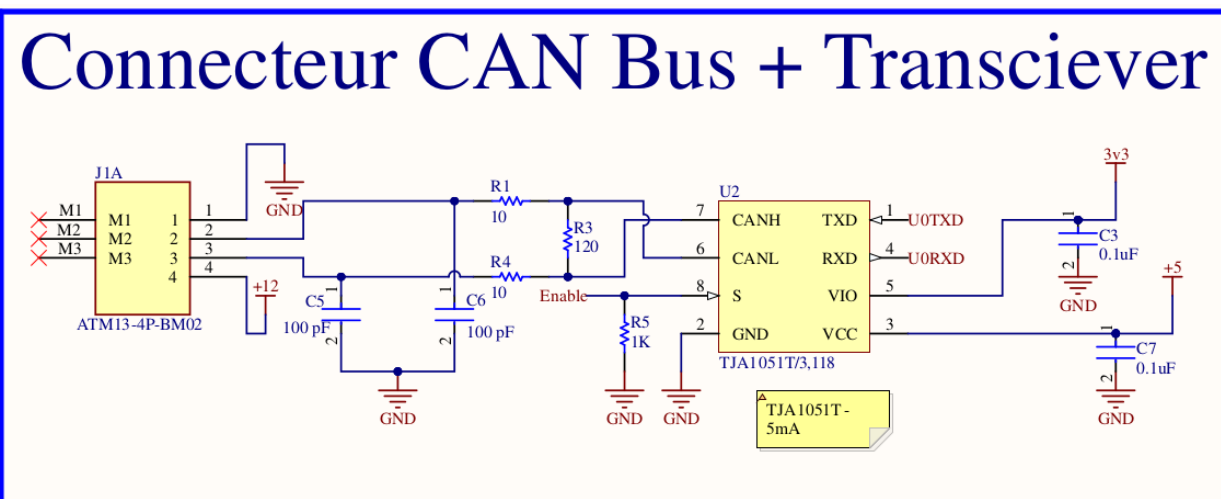
Le schéma électrique complet est dans l'annexe. **Annexe 1 : Schémas électrique**

### 4.1 Régulateur de tension



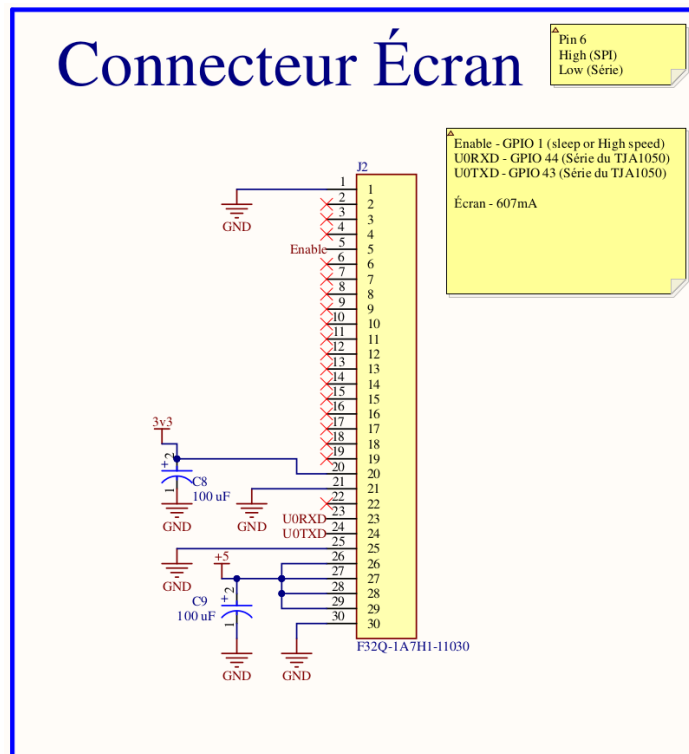
Le circuit d'alimentation est composé d'un LM2675M-5.0 qui est un régulateur de découpage. Ce régulateur prend une tension de 12V en entrée et donne une tension de 5V en sortie et fournie 1A. Une DEL verte sert à indiquer qu'il y a une tension de 5V en sortie.

### 4.2 Connecteur CAN Bus et Transmetteur



Ce circuit est la section du réseau CAN et du transmetteur. L'alimentation et le CAN arrivent du connecteur 4 pins. Le réseau CAN va au transmetteur (TJA1051T/3, 118). Ce transmetteur reçoit une tension de 3v3 sur la patte 5 pour aider la communication entre l'écran et le TJA1051T. La patte 8 définit si le transmetteur transmet ou reçoit. Les pattes 1 et 4 sont du série CAN. Elle envoie à l'écran la trame CAN.

### 4.3 Connecteur de l'écran

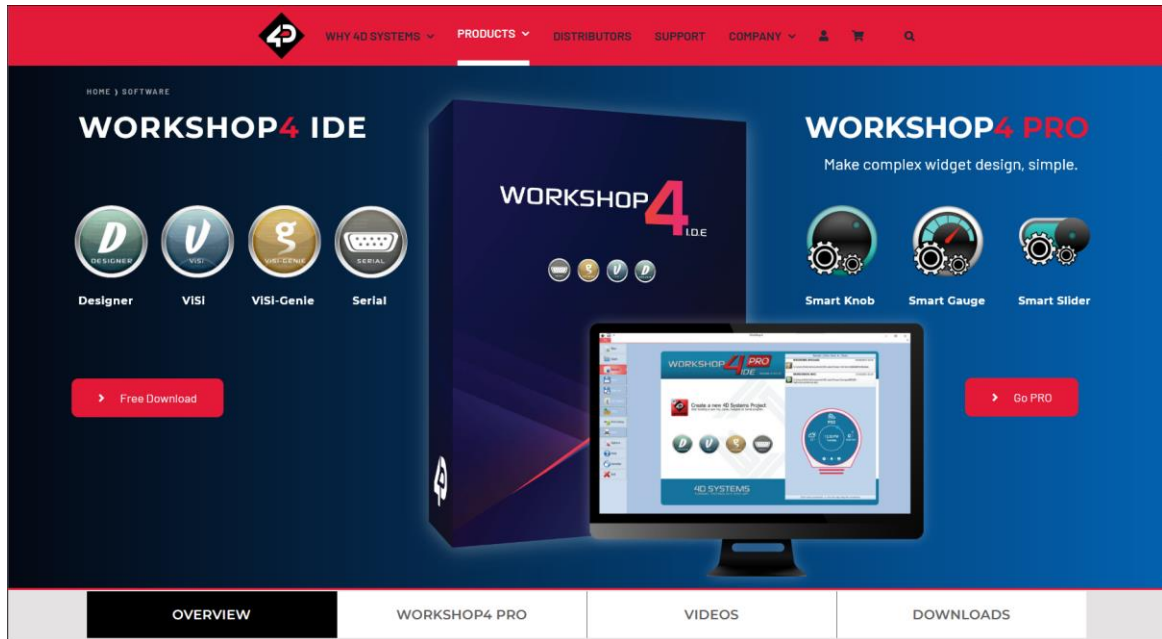


Le connecteur F32Q-1A7H1-11030 fait la liaison entre le PCB et l'écran de 4D Systems. Il alimente en 5V l'écran sur 4 pins. Elle fournit aussi le 3v3 au TJA1051T sur la patte Vio. La patte 5 est le GPIO1. Elle envoie un niveau haut ou bas pour modifier le TJA1051T en émetteur ou receveur. Nous la mettons à un niveau bas pour que le transmetteur soit en receveur.

## 5 Contenu logiciel

### 5.1 Logiciel Utilisé

Le logiciel que nous utilisons est Workshop4 IDE. Ce logiciel est celui de la compagnie 4D System. Voici le lien pour l'installation : <https://4dsystems.com.au/software/>.



Workshop4 est un logiciel qui fonctionne avec Arduino, cela signifie que nous devons coder en C++. Il nous permet de mettre des widgets sur l'affichage.

### 5.2 Librairie utilisé

Il y a trois librairies que nous utilisons.

```
#include "gfx4desp32_%%displaynm%%.h"
#include "Code_Final_CANVision_V1_FACConst.h"
#include <ESP32-TWAI-CAN.hpp>
```

La première librairie est seulement pour définir la dimension de l'écran.

La deuxième librairie est pour définir les widgets du code au complet.

La troisième librairie est pour le CAN, soit la plus importante. Pour l'utilisation du projet, nous utilisons seulement la partie de lecture de la trame CAN. Voici le créateur de cette librairie : <https://github.com/handmade0octopus/ESP32-TWAI-CAN>.

### 5.3 Compréhension du code principale

Tout le contenu logiciel pour utiliser le projet peut être retrouvé sur le GitHub du projet.

[https://github.com/FelixDuck15/projet\\_Formule\\_SAE/tree/main](https://github.com/FelixDuck15/projet_Formule_SAE/tree/main)

Essentiellement, tout le code réside dans le dossier projet\_Formule\_SAE/H24/LOGICIEL/AFFICHAGE\_SAE/Code\_Final\_CANVision\_V1\_FA.

Le code de la boucle principale est le suivant :

```
void loop()
{
    btn(); // verifie les boutons s'il sont appuyer
    readCAN(); // lis le BUS CAN
    switch(pageValue){
        case 1:
            hv_page(false); // appel la page HV
            break;
        case 2:
            temp_page(false); // appel la page Temp
            break;
        case 3:
            lv_page(false); // appel la page LV
            break;
    }
}
```

Ce code vérifie tout le temps si un bouton de l'écran a été touché. Si un de ces boutons a été touché, il va nous diriger vers la page demandée. La fonction « readCAN() » va lire les trames reçues et afficher les bonnes valeurs sur la page active.

## 6 Procédure de développement

### 6.1 Téléverser dans l'écran

- 1) Allez sur l'onglet « Comms » et vérifiez s'il s'agit du bon port COM.
- 2) Allez sur l'onglet « Home » et appuyez sur « Compile ». Il va automatiquement vérifier tous les widgets et avoir une barre de progression qui va créer les fichiers des widgets qui sont des extensions « dat » et « gci ». Cliquez sur « No Thanks ».
- 3) Vérifiez qu'il n'y a pas d'erreur.
- 4) Quand tout est compilé et sans erreur, cliquez sur « Download ».
- 5) Prenez la carte SD à l'arrière de l'écran et mettez les fichiers « dat » et « gci » qui ont été créés à l'étape 2. Remettez la carte SD dans l'écran.
- 6) Alimentez l'écran et il devrait faire le reste automatiquement.

### 6.2 Installation des logiciels et des librairies

- 1) Installez Arduino IDE 2.0 <https://www.arduino.cc/en/software>  
Downloads



**Arduino IDE 2.3.2**

The new major release of the Arduino IDE is faster and even more powerful! In addition to a more modern editor and a more responsive interface it features autocompletion, code navigation, and even a live debugger.

For more details, please refer to the [Arduino IDE 2.0 documentation](#).

Nightly builds with the latest bugfixes are available through the section below.

**SOURCE CODE**

The Arduino IDE 2.0 is open source and its source code is hosted on [GitHub](#).

**DOWNLOAD OPTIONS**

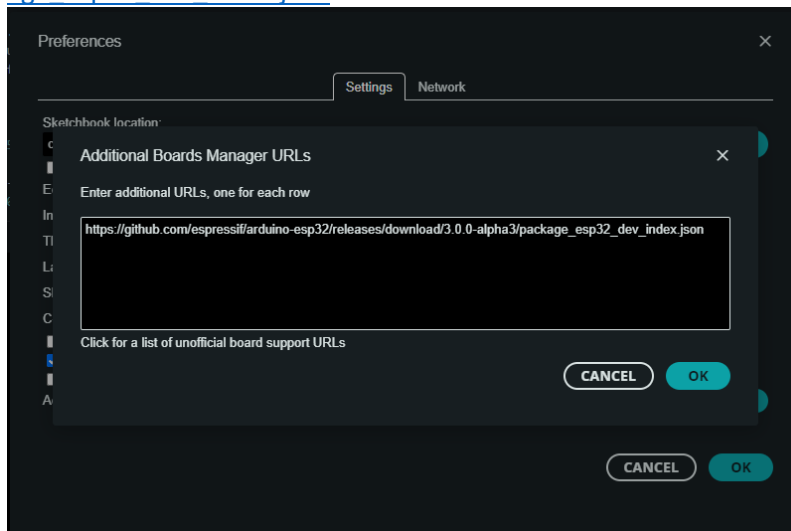
**Windows** Win 10 and newer, 64 bits  
**Windows** MSI installer  
**Windows** ZIP file  
**Linux** Applimage 64 bits (X86-64)  
**Linux** ZIP file 64 bits (X86-64)  
**macOS** Intel, 10.15: "Catalina" or newer, 64 bits  
**macOS** Apple Silicon, 11: "Big Sur" or newer, 64 bits

[Release Notes](#)

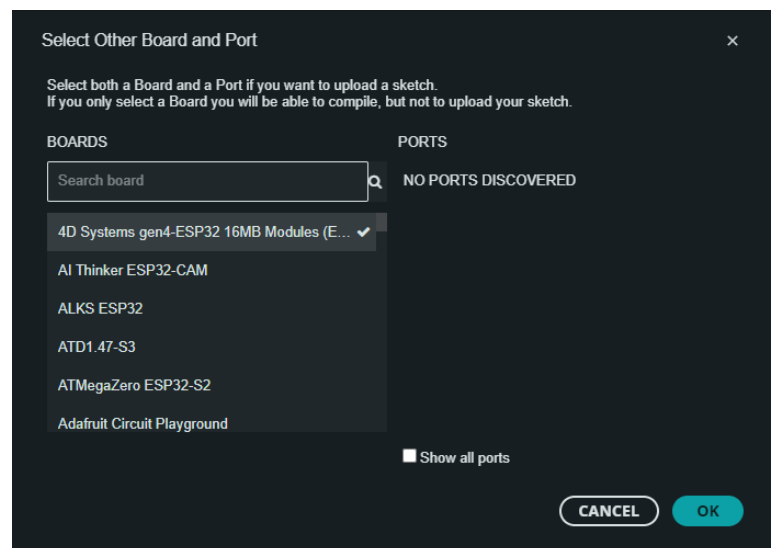
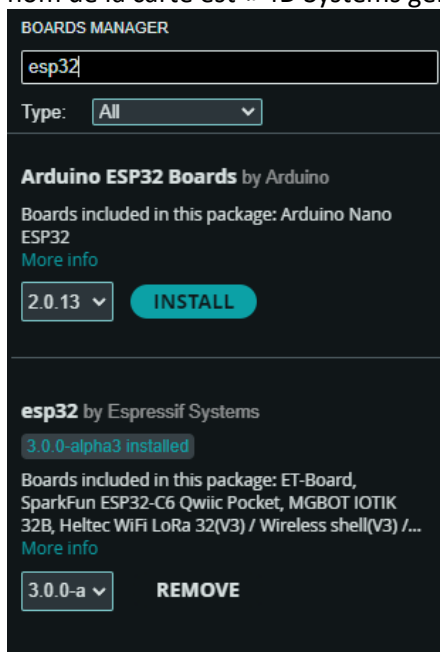
- 2) Installez Workshope4 IDE (seulement sur Windows) <https://4dsystems.com.au/software/>

- 3) Dans « Fichier/Préférence » d'Arduino, collez la référence suivante pour avoir les cartes ESP32 dans « URL de gestionnaire de cartes supplémentaires » :

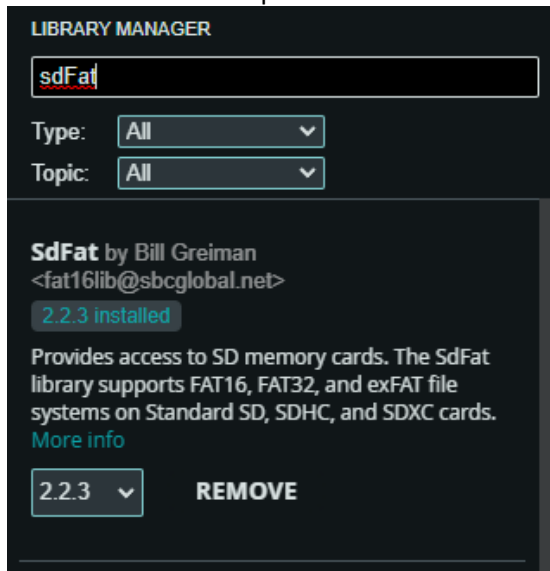
[https://github.com/espressif/arduino-esp32/releases/download/3.0.0-alpha3/package\\_esp32\\_dev\\_index.json](https://github.com/espressif/arduino-esp32/releases/download/3.0.0-alpha3/package_esp32_dev_index.json)



- 4) Dans « Gestionnaire de cartes », installez « esp32 » version 3.0.0-a par Espressif Systems. Le nom de la carte est « 4D Systems gen4-ESP32 16MB Modules (ESP32-S3R8n16) ».



- 5) Installez la version la plus récente de la librairie « sdFat ».

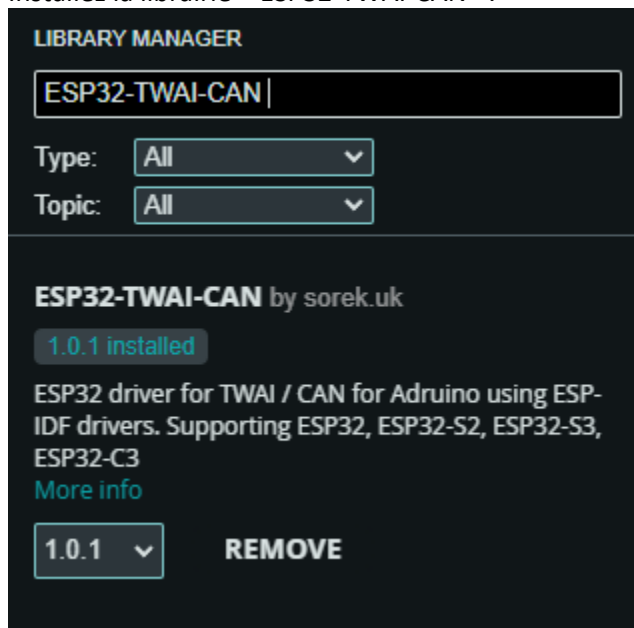


- 6) Téléchargez puis installez la librairie « GFX4dESP32 » via ce lien : <https://github.com/4dsys-tems/GFX4dESP32/tree/arduino-esp32-v3>





- 7) Installez la librairie « ESP32-TWAI-CAN ».



Vous êtes maintenant prêts à utiliser Workshope4 IDE. Le fichier de code utilisé par Workshope4 IDE est celui avec l'extension « 4dArdUsd ».

### 6.3 Modifier l'affichage du projet

- 1) Le code de 4D system est le fichier avec l'extension « 4dArdUsd ».
- 2) Pour modifier un widget, il faut cliquer dessus et vérifier le nom dans « object » s'il s'agit du bon widget. Changer les propriétés dans « Properties » ou dans le code. S'il est changé dans « Properties », il faut appuyer sur « Paste Static Code ».
- 3) Pour ajouter un widget ou une page, il faut aller dans l'onglet « Widget ». Il faut choisir entre : « Background », « Buttons », « Digits », « Gauges », « Inputs », « Labels », « Primitives », « System/Media ».

## 6.4 Modifier le code CAN

Il y a deux parties du code CAN: void setup () et void readCAN (). Pour la partie « void setup () », le code ressemble à ceci :

```
// CAN
ESP32Can.setPins(CAN_TX, CAN_RX); // initialise les pin série CAN
ESP32Can.setSpeed(ESP32Can.convertSpeed(1000)); // initialise la vitesse de CAN
ESP32Can.setRxQueueSize(1); // Initialisation du queue size
ESP32Can.setTxQueueSize(1);

[...]

// Debut du CAN
if(ESP32Can.begin()){ // si le CAN est bien initialiser
    gfx.println("CAN bus started!");
    delay(1000);
}
else // si pas bien initialiser
{
    noDataRecieved();
    while(1);
}
```

Ce code contient les pins du série CAN, la vitesse de transmission ainsi que les queues de la réception et de la transmission du série CAN. Aussi, le programme vérifie si le CAN est actif ou non. Si le CAN est actif, un message de confirmation s’affiche sur l’écran. Sinon, une fonction affiche que le programme ne reçoit rien et il entre dans une boucle infinie.

Pour la partie « void readCAN () » :

```
void readCAN(){
  if(ESP32Can.readFrame(rxFrame, 50))      // lecture à chaque 50ms
  {
    idCAN1 = rxFrame.identifrier;          // ID
    canTest1 = rxFrame.data[0];            // centaine des données
    canTest2 = rxFrame.data[1];            // unites des données
    totalCAN = (canTest1*100) + canTest2;    // faire un chiffre entre 92 et 425
    if(totalCAN == 92){                     // si on reçoit le minimum du capteur
      dataArray[1] = 0;
    }

    else if(totalCAN > 92 && totalCAN < 425){
      dataArray[1] = map(totalCAN, 92, 425, 0, 100);
    }
  }

  else if(totalCAN == 425){                 // si on reçoit le maximum du capteur
    dataArray[1] = 100;
  }
}
```

Ce code est encore en développement, car il faudrait 10 parties au total. Soit, une partie pour chaque widget. Cependant, ce code fait fonctionner le widget « SOH » de la page HV. Elle vérifie l'identifiant envoyé et prend les deux données. La donnée rxFrame.data[0] est une valeur de 0 à 9 et rxFrame.data[1] est une valeur entre 0 et 99. La fonction « map » sert à prendre les valeurs reçues et les mettre dans les limites du widget.

7 Listes de matériel et coûts

Nom et numéro de la pièce	Description	Lien Web chez le fournisseur	Prix unitaire	Quantité	Transport	Total	Sans écran
gen4-ESP32-70CT Starter Kit	L'écran de 4D System	<a href="https://4dsystems.com.au/products/gen4-esp32-70ct/#Description">https://4dsystems.com.au/products/gen4-esp32-70ct/#Description</a>	220,05 \$	1		220,05 \$	
ATM13-4P	Gros connecteur CAN 4Pin	<a href="https://www.digikey.ca/en/products/detail/amphenol-sine-systems-corp/ATM13-4P-BM02/9836883">https://www.digikey.ca/en/products/detail/amphenol-sine-systems-corp/ATM13-4P-BM02/9836883</a>	14,74 \$	1		14,74 \$	14,74 \$
PCB	PCB	<a href="https://cart.jlpcb.com/quote?orderType=1&amp;homeUplloadNum=07dbbcc42247433e96adc198a3f9c333&amp;businessType=example&amp;fileNameGerber_file_AffichageCAN_F.A.GUIMONT.zip">https://cart.jlpcb.com/quote?orderType=1&amp;homeUplloadNum=07dbbcc42247433e96adc198a3f9c333&amp;businessType=example&amp;fileNameGerber_file_AffichageCAN_F.A.GUIMONT.zip</a>	2,00 \$	1	50	52,00 \$	2,00 \$
TJA1051T/3,118	transmetteur de CAN	<a href="https://www.digikey.ca/fr/products/detail/microchip-technology/MCP2515T-I-SO/593681/">https://www.digikey.ca/fr/products/detail/microchip-technology/MCP2515T-I-SO/593681/</a>	3,78 \$	1		3,78 \$	3,78 \$
LM2675M-5.0/NOPB	Régulateur de tension 12V à 5V-1A	<a href="https://www.digikey.ca/en/products/detail/texas-instruments/LM2675M-5-0-NOPB/271162">https://www.digikey.ca/en/products/detail/texas-instruments/LM2675M-5-0-NOPB/271162</a>	8,75 \$	1		8,75 \$	8,75 \$
2073-05-30-D-0030-A-4-06-4-T-ND	CABLE FFC/FPC 30 pin 0.5mm	<a href="https://www.digikey.ca/en/products/detail/gct/05-30-D-0030-A-4-06-4-T/21266133">https://www.digikey.ca/en/products/detail/gct/05-30-D-0030-A-4-06-4-T/21266133</a>	1,98 \$	1		1,98 \$	1,98 \$
boitier	boitier imprimer en 3D	<a href="https://cad.onshape.com/documents/03c94472e16741a177019661/w/f4f8831b2ce90ca943e7f51e/e/c7fe7dde4c521339c3673f8e">https://cad.onshape.com/documents/03c94472e16741a177019661/w/f4f8831b2ce90ca943e7f51e/e/c7fe7dde4c521339c3673f8e</a>	1,00 \$	1		1,00 \$	1,00 \$
Pieces discrètes	Les pieces discrètes	Lien dans la page "Pieces discrètes"				10,31 \$	10,31 \$

Sous-total	312,61 \$	42,56 \$
Taxes	46,89 \$	6,38 \$
Grand Total	359,50 \$	48,94 \$

Figure 8: liste de matériaux

Nom et numéro de la pièce	Description	Lien Web chez le fournisseur	Prix unitaire	Quantité	Total
CAP ALUM 100UF 20% 50V SMD / EEE-FH1H101L	100 µF 50 V Aluminum Electrolytic Capacitors Radial, Can - SMD 500mOhm @ 100kHz 10000 Hrs @ 105°C	<a href="https://www.digikey.ca/en/products/detail/panasonic-electronic-components/EEE-FH1H101L/20372560?s=N4lgTCBcDaIKIIGIAkCMaAMqAylC6AvkA">https://www.digikey.ca/en/products/detail/panasonic-electronic-components/EEE-FH1H101L/20372560?s=N4lgTCBcDaIKIIGIAkCMaAMqAylC6AvkA</a>	1,63	3	4,89 \$
condensateur 68uF / EEE-FN1H680XP	CAP ALUM 68UF 20% 50V SMD	<a href="https://www.digikey.ca/en/products/detail/panasonic-electronic-components/EEE-FN1H680XP/11656970">https://www.digikey.ca/en/products/detail/panasonic-electronic-components/EEE-FN1H680XP/11656970</a>	1,03	1	1,03 \$
F32Q-1A7H1-11030	Flex Connector, 0.50mm Pitch, Height 2.00mm, Right angle, Slider type, ZIF, Upper contact, 30 position , Without MYLAR	<a href="https://www.digikey.ca/en/products/detail/amphenol-cs-fci/F32Q-1A7H1-11030/11564712">https://www.digikey.ca/en/products/detail/amphenol-cs-fci/F32Q-1A7H1-11030/11564712</a>	1,38	1	1,38 \$
condensateur 10 nF / 80-C0603C154Z3V	Chip Capacitor, 10 nF, +/- 10%, 50 V, -55 to 125 degC, 0603 (1608 Metric), RoHS, Tape and Reel	<a href="https://www.mouser.ca/ProductDetail/KE-MET/C0603C154Z3VACTU?qs=dIXU2iuYu4SgHNv%2Fd0eG3Q%3D%3D">https://www.mouser.ca/ProductDetail/KE-MET/C0603C154Z3VACTU?qs=dIXU2iuYu4SgHNv%2Fd0eG3Q%3D%3D</a>	0,41	1	0,41 \$
condensateur 100 nF / CL21B104KCFNNNE	Automotive Ceramic Capacitor, 0805, 100nF, 10%, X7R, 15%, 100V	<a href="https://www.digikey.ca/en/products/detail/samsung-electro-mechanics/CL21B104KCFNNNE/5961324">https://www.digikey.ca/en/products/detail/samsung-electro-mechanics/CL21B104KCFNNNE/5961324</a>	0,15	2	0,30 \$
condensateur 100pF / C0805C101J5GAC7800	CAP CER 100PF 50V C0G/NPO 0805	<a href="https://www.digikey.ca/en/products/detail/kemet/C0805C101J5GAC7800/411121">https://www.digikey.ca/en/products/detail/kemet/C0805C101J5GAC7800/411121</a>	0,15	2	0,30 \$
LED verte SMD / LG R971-KN-1-0-20-R18	LED GREEN CLEAR 0805 SMD	<a href="https://www.digikey.ca/en/products/detail/ams-osram-usa-inc/LG-R971-KN-1-0-20-R18/1227925">https://www.digikey.ca/en/products/detail/ams-osram-usa-inc/LG-R971-KN-1-0-20-R18/1227925</a>	0,42	1	0,42 \$

SS14FL	DIODE SCHOTTKY 40V 1A SOD123F	<a href="https://www.digikey.ca/en/products/detail/onsemi/SS14FL/5305065?0=%2Frectifiers%2Fsingle-diodes%2Fsod-123f&amp;s=N4IgiCBcpgbFoDGUBmBDANgZwKYBoQA3AOygBcAnAV3xAHsoBtEAFgAY-WAOAZgCYQAugQAOZKCADKIAJ-bEASiAC%2BBAKwB2NghDJI6bLQaRm7NtzWdBIsZEKz5SggE4wLLTr24Ch5mDCO1KppCIKLiUhSyCsog-vGxqENDaqJie9EysbAAEAGqWldYgAKrE0mQA8igAsjhoWFQUOErR-EaSEi4AYgAygopAA">https://www.digikey.ca/en/products/detail/onsemi/SS14FL/5305065?0=%2Frectifiers%2Fsingle-diodes%2Fsod-123f&amp;s=N4IgiCBcpgbFoDGUBmBDANgZwKYBoQA3AOygBcAnAV3xAHsoBtEAFgAY-WAOAZgCYQAugQAOZKCADKIAJ-bEASiAC%2BBAKwB2NghDJI6bLQaRm7NtzWdBIsZEKz5SggE4wLLTr24Ch5mDCO1KppCIKLiUhSyCsog-vGxqENDaqJie9EysbAAEAGqWldYgAKrE0mQA8igAsjhoWFQUOErR-EaSEi4AYgAygopAA</a>	0,38	1	0,38 \$
résistance 1K / RC0805FR-071KL	Chip Resistor, 1 KOhm, +/- 1%, 125 mW, -55 to 155 degC, 0805 (2012 Metric)	<a href="https://www.digikey.ca/en/products/detail/yageo/RC0805FR-071KL/727444">https://www.digikey.ca/en/products/detail/yageo/RC0805FR-071KL/727444</a>	0,15	1	0,15 \$
Bobine 47uH / NPI31W470MTRF	Inductor Power Unshielded Wire- wound 47uH 20% 100KHz 1.8A 0.14Ohm DCR T/R	<a href="https://www.arrow.com/en/products/npi31w470mtrf/nic-components">https://www.arrow.com/en/products/npi31w470mtrf/nic-components</a>	0,45	1	0,45 \$
résistance 270 hom SMD / RC0805JR- 07270RL	RES 270 OHM 5% 1/8W 0805	<a href="https://www.digikey.ca/en/products/detail/yageo/RC0805JR-07270RL/728291">https://www.digikey.ca/en/products/detail/yageo/RC0805JR-07270RL/728291</a>	0,15	1	0,15 \$
résistance 120 hom SMD / RC0805JR- 07120RL	RES 120 OHM 5% 1/8W 0805	<a href="https://www.digikey.ca/en/products/detail/yageo/RC0805JR-07120RL/728247">https://www.digikey.ca/en/products/detail/yageo/RC0805JR-07120RL/728247</a>	0,15	1	0,15 \$
résistance 10 hom SMD / RMCF0805JG10R0	RES 10 OHM 5% 1/8W 0805	<a href="https://www.digikey.ca/en/products/detail/stackpole-electronics-inc/RMCF0805JG10R0/1711843">https://www.digikey.ca/en/products/detail/stackpole-electronics-inc/RMCF0805JG10R0/1711843</a>	0,15	2	0,30 \$

Sous- total	10,31 \$
Taxes	1,55 \$
Grand Total	11,86 \$

## 8 Modifications et améliorations

- 1) Il serait préférable de sceller le boîtier pour ne pas avoir aucune entrée d'eau. Peut-être utiliser une gaine en caoutchouc qui se colle sur le devant du boîtier pour qu'il y aille cet aspect de sceller.
- 2) Il faudrait avoir un plus petit boîtier, car il est très épais à l'arrière. Peut-être, le réduire de 3 ou 4 centimètres.
- 3) Utiliser un autre connecteur que ATM13-4P parce qu'il est trop gros. Peut-être utiliser un connecteur fait en long donc les pins une par-dessus l'autre.

## 9 Annexe

### 9.1 Annexe 1 : Schémas électriques

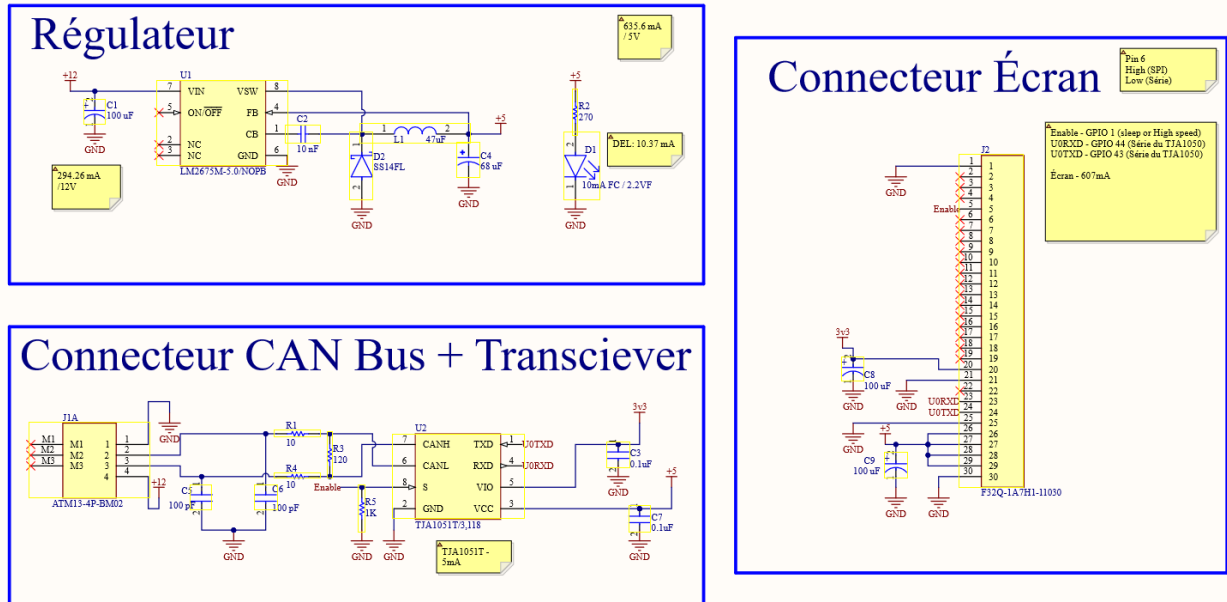


Figure 9: Schémas électriques complets