

Académie de Besançon

BACCALAURÉAT SCIENCES ET TECHNOLOGIES
INDUSTRIELLES

Spécialité génie électronique

Session 2010

Épreuve de construction électronique

THÈME :

Gestion éolienne et Domotique

Lycée Cournot

Gray

Sommaire

1.	Mise en situation	4
1.1.	Introduction	4
1.1.1.	Fonctionnement d'une éolienne	4
1.1.2.	Fonctionnement d'un panneau solaire	5
1.1.3.	Gestion éolienne	5
1.1.4.	Domotique	5
1.1.5.	Serveur Web embarqué	6
1.2.	Éléments constitutifs du système technique	6
2.	Description fonctionnelle du système technique	7
2.2.	Définition des liaisons	8
3.	Analyse fonctionnelle	8
3.1.	Fonction globale	8
3.2.	Elargissement de l'étude	8
3.3.	Schéma fonctionnel niveau II	8
3.4.	Fonction d'usage	8
3.5.	Etude fonctionnelle de degré 1	8
3.5.1.	Schéma fonctionnel de degré 1 de OT1 et OT2	9
3.5.2.	Définition des fonctions principales	10
3.6.	Etude fonctionnelle de degré 2	11
3.6.1.	Schéma fonctionnel de degré 2 de FP10	11
3.6.2.	Définition des fonctions secondaires de FP10	11
3.6.3.	Schéma fonctionnel de degré 2 de FP11	12
3.6.4.	Définition des fonctions secondaires de FP11	12
3.6.5.	Schéma fonctionnel de degré 2 de FP20	14
3.6.6.	Définition des fonctions secondaires de FP20	14
3.6.7.	Schéma fonctionnel de degré 2 de FP21	14
3.6.8.	Définition des fonctions secondaires de FP21	14
3.6.9.	Schéma fonctionnel de degré 2 de FP22	15
3.6.10.	Définition des fonctions secondaires de FP22	15
3.6.11.	Schéma fonctionnel de degré 2 de FP23	15
3.6.12.	Définition des fonctions secondaires de FP23	16
3.6.13.	Schéma fonctionnel de degré 2 de FP3	16
3.6.14.	Définition des fonctions secondaires de FP3	17
3.6.15.	Schéma fonctionnel de degré 2 de FP4	19
3.6.16.	Définition des fonctions secondaires de FP4	19
3.6.17.	Schéma fonctionnel de degré 2 de FP5	20
3.6.18.	Définition des fonctions secondaires de FP5	20
3.6.19.	Schéma fonctionnel de degré 2 de FP6	22
3.6.20.	Définition des fonctions secondaires de FP6	22
3.6.21.	Schéma fonctionnel de degré 2 de FP7	23
3.6.22.	Définition des fonctions secondaires de FP7	23
3.6.23.	Schéma fonctionnel de degré 2 de FP8	24
3.6.24.	Définition des fonctions secondaires de FP8	24
4.	Schémas structurels	25
4.1.	FP10	26
4.2.	FP11	27
4.3.	FP21	30
4.4.	FP22	31
4.5.	FP3	32
4.6.	FP4	36

4.7.	FP5	37
4.8.	FP6	38
4.9.	FP7	39
4.10.	FP8	40
4.11.	Nomenclature	41
5.	TRAVAUX ELEVES	44
5.1.	Travail groupe 1	44
5.2.	Travail groupe 2	45
5.3.	Travail groupe 3	46
5.4.	Travail groupe 4	47
5.5.	Travail groupe 5	47
5.6.	Travail groupe 6	49
5.7.	Plan du rapport :	50
6.	DEFINITION DES FONCTIONS LOGICIELLES FOURNIES	51
6.1.	Fichiers à inclure	51
6.2.	Afficheur LCD	51
6.3.	Bus I2C	53
7.	COMPOSANTS PARTICULIERS.....	53
8.	LOGICIELS UTILISES.....	53

1. Mise en situation

1.1. Introduction

Depuis 150 ans, la révolution industrielle a créé de vastes richesses et a considérablement amélioré les conditions de vie humaine dans les pays industrialisés. Le moteur de cette révolution a été le charbon puis le pétrole qui ont permis de multiplier la productivité du travail humain. Bien que nous ayons tiré beaucoup de bénéfices de cette intensive utilisation de l'énergie, nous sommes face maintenant à 2 problèmes importants:

- le réchauffement du climat.
- la raréfaction des énergies fossiles.

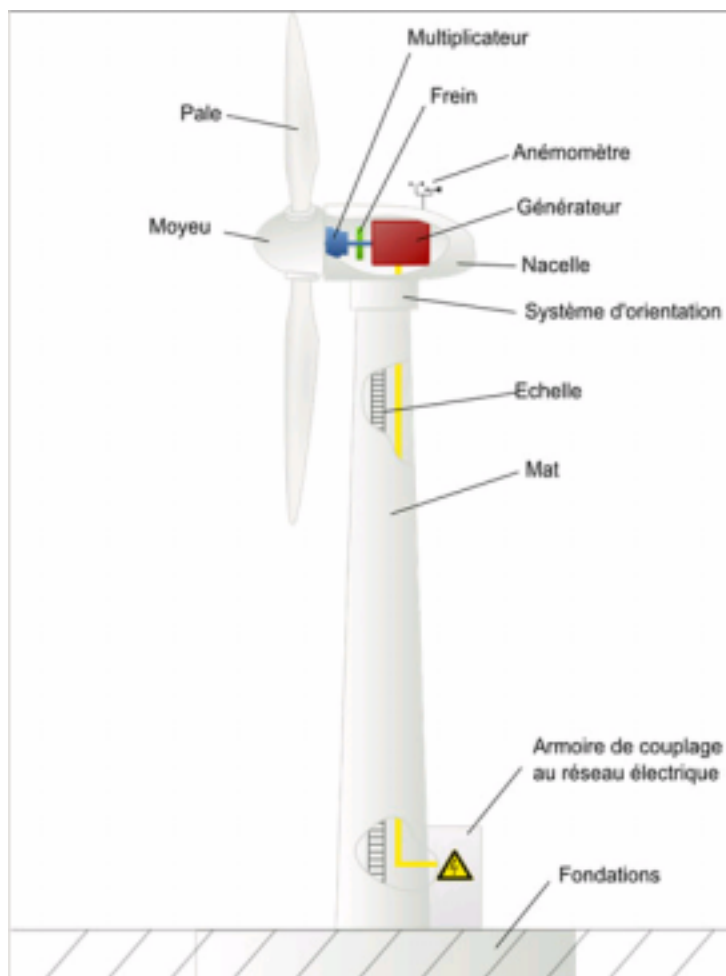
Il apparaît que les sociétés du futur pourraient dépendre de sources d'énergies petites, diversifiées et renouvelables de type éolienne, solaire, biodiversités... Largement dispersés sur tout le territoire, ces centres de production d'énergie auraient le double avantage de fournir de l'énergie sans émission de gaz à effet de serre et d'être moins dépendant des nations productrices de pétrole. C'est pourquoi de nombreuses entreprises proposent maintenant des installations éoliennes et/ou solaires de faibles ou moyennes puissances pouvant être installées chez des particuliers. C'est le cas du système décrit dans ce dossier.

Le système est composé de 2 sources d'alimentation en énergie: une éolienne et un panneau solaire. L'énergie éolienne et l'énergie solaire sont transformées en énergie électrique qui est ensuite stockée dans une batterie. Un système de surveillance donne toutes les informations disponibles à l'utilisateur qui pourra localement ou à distance connecter ou déconnecter des charges.

Afin d'optimiser le rendement de l'éolienne, il importe de placer celle-ci dans un endroit le plus soumis aux vents. Dans ce but une mesure de la vitesse et de l'orientation du vent doit être faite avant toute implantation définitive d'éolienne.

Par ailleurs une partie domotique permet de surveiller l'habitation (incendie et intrusion)

1.1.1. Fonctionnement d'une éolienne



Une éolienne capte l'énergie cinétique du vent (les pales tournent) et la convertit en un couple qui fait tourner le rotor (couplé sur l'arbre des pales). Le rotor entraîne en rotation un aimant créant ainsi un champ magnétique tournant. Les bobines du stator, soumises à ce champ tournant vont alors créer un courant électrique de fréquence proportionnelle à la vitesse de rotation du rotor.

Elle fonctionne uniquement lorsque la vitesse du vent est comprise entre 15 km/h et 90 Km/h maximum, pour des raisons de sécurité. La puissance d'une éolienne peut aller jusqu'à 5 MW (A titre de comparaison, la puissance d'une tranche de centrale nucléaire est de 1400 MW environ). L'électricité produite est acheminée par un câble électrique souterrain jusqu'au poste de raccordement. Au pied de chaque éolienne, un transformateur convertit la tension de 690 volts en 20000 volts, tension du réseau national d'Électricité de France sur lequel toute l'électricité produite est acheminée.

1.1.2. Fonctionnement d'un panneau solaire



QU'EST CE QUE L'EFFET PHOTOVOLTAÏQUE ?

L'énergie solaire photovoltaïque (à distinguer de l'énergie solaire thermique) provient de la conversion de la lumière du soleil en électricité. Cette conversion se produit au sein de matériaux "semi-conducteurs", qui ont comme propriété de libérer leurs électrons sous l'influence d'une énergie extérieure. Dans le cas du photovoltaïque, cette énergie est apportée par les photons, les composants de la lumière, qui heurtent les électrons et les libèrent, induisant le courant électrique.

QU'EST CE QU'UNE CELLULE PHOTOVOLTAÏQUE ?

Les cellules photovoltaïques ont pour constituant de base les semi-conducteurs, dont le plus exploité est le silicium. Ce matériau est modifié par l'apport d'éléments extérieurs (en général des atomes de phosphore et de bore) pour le polariser et attirer les électrons dans un certain sens. Orientés de la sorte par le matériau "dopé", ils vont circuler et créer un courant électrique lui-même recueilli par des fils métalliques très fins. Les cellules sont assemblées en modules. Les modules vont de micro puissances de moins de 2 W à des capacités unitaires de 150 W.

1.1.3. Gestion éolienne

Une gestion des paramètres physiques de l'éolienne (**orientation et vitesse du vent**) est réalisée afin de pouvoir alimenter une base de données. En effet l'implantation optimale d'une éolienne est réalisée après une étude des vents en terme de force, d'orientation et de fréquence (mesure de la durée des périodes sans vent).

1.1.4. Domotique

La domotique comprend l'assistance électronique et informatique à la gestion d'une habitation en terme de gestion énergie (chauffage...) sécurité (incendie) protection (intrusion, vol)

La partie domotique de l'étude comprend : **Alerte incendie et Alerte intrusion**

L'alerte incendie est déclenchée en cas de dépassement de température critique ou de présence de fumée à l'intérieur d'un local ou d'une pièce d'habitation.

L'alerte intrusion est activée lorsqu'une personne est détectée dans l'habitation et qu'elle n'a pas entré le code d'accès dans le délais impartit.

1.1.5. Serveur Web embarqué

Tous les paramètres énoncés ci-dessus sont résumés dans une page HTML hébergée dans un serveur web embarqué :

- paramètres vitesse et direction du vent
- alerte incendie (message d'alerte)
- alerte intrusion (message d'alerte)
- commande à distance de la commutation sur batterie et allumage d'éclairage
- avec en plus une surveillance de l'état de la batterie sous forme de témoin (vert orange rouge ou noir)

The screenshot shows a Mozilla Firefox browser window with the URL <http://10.100.241.226>. The page title is "Gestion Eolienne et Domotique". The interface consists of several panels:

- Temoin charge batterie:** A green square indicator.
- Paramètres éolienne:** A panel with a compass icon showing "Vitesse vent: 10 km/h" and "Direction vent: 130°".
- Commandes sorties:** A panel with "Lumière = ", "Commutation batterie = ", and an "Envoi" button.
- [Application JavaScript]:** A dialog box with a yellow warning icon and the text "ALERTE INCENDIE", with an "OK" button.

At the bottom of the page, it says "Le 09/10/2009 à 13:08:33" and "Thème 2010 - Section STI Lycée Cournot 70100 GRAY".

1.2. **Éléments constitutifs du système technique**

Eolienne,

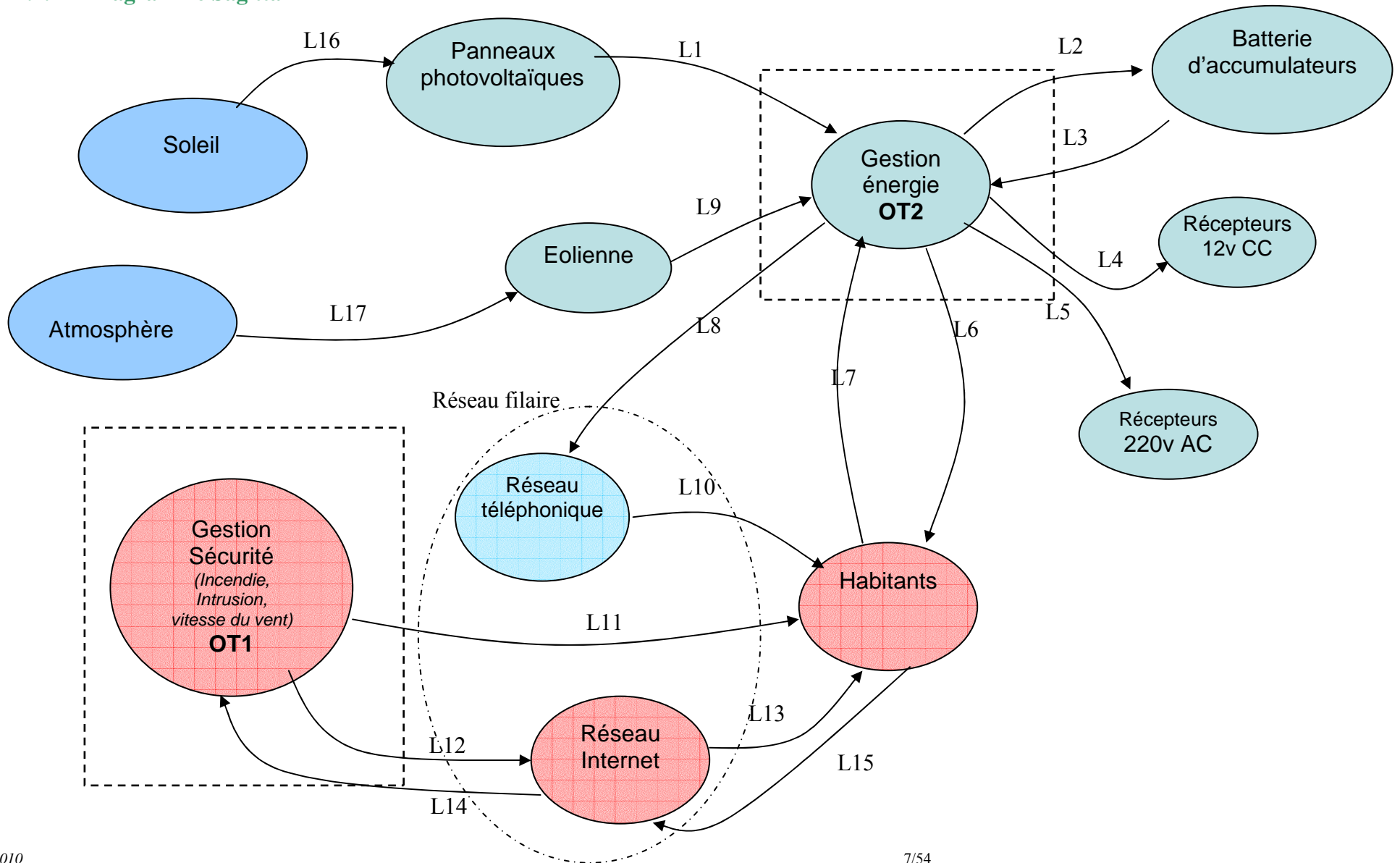
Panneaux solaires, batterie

Gestion Electronique : regroupe l'ensemble des cartes à étudier.

PC : Le PC est relié au réseau ethernet. L'utilisateur peut consulter les paramètres domotiques et énergie en consultant le serveur WEB embarqué dans la partie **Gestion Electronique**. Il suffit de taper l'adresse IP (par exemple [http:// 10.100.241.226](http://10.100.241.226) dans la barre d'un navigateur Internet (Mozilla, IE))

2. Description fonctionnelle du système technique

2.1. Diagramme Sagittal



2.2. Définition des liaisons

- L1: Energie électrique: tension unipolaire.
- L2: Energie électrique: tension unipolaire.
- L3: Energie électrique: tension unipolaire.
- L4: Energie électrique: tension unipolaire.
- L5: Energie électrique: tension sinusoïdale.
- L6: Informations visuelles (état de la batterie)
- L7: Information de commande (connections ou déconnection de la charge)
- L8: Information électrique (alerte : tension de batterie trop faible)
- L9: Energie électrique: tension unipolaire.
- L10: Information sonore (alarme : tension de batterie trop faible)
- L11: Information sonore (alarme : intrusion, incendie)
- L12: Information électrique (vitesse et orientation du vent, alarme intrusion, incendie)
- L13: Information visuelle (vitesse et orientation du vent, alarme intrusion, incendie)
- L14: Information électrique (commande commutation batterie, allumage lumière)
- L15: Information manuelle (commande commutation batterie, allumage lumière)
- L16: Energie solaire
- L17: Energie éolienne.

3. Analyse fonctionnelle

3.1. Fonction globale

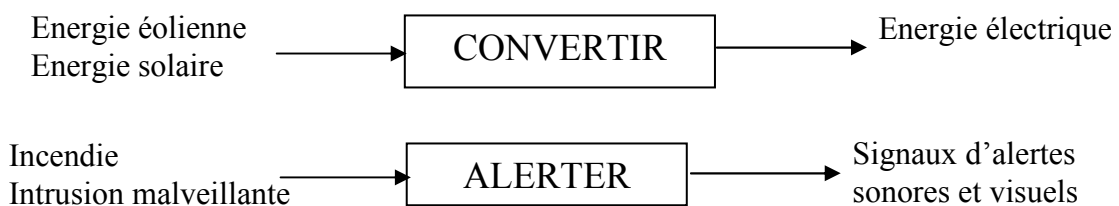
Transformer une énergie en électricité exploitable.
Gérer la sécurité d'une habitation

3.2. Elargissement de l'étude

Autres systèmes ayant la même fonction globale :

- les centrales hydrauliques
- les centrales thermiques
- les centrales nucléaires
- Les centrales géothermiques

3.3. Schéma fonctionnel niveau II



3.4. Fonction d'usage

OT1 :

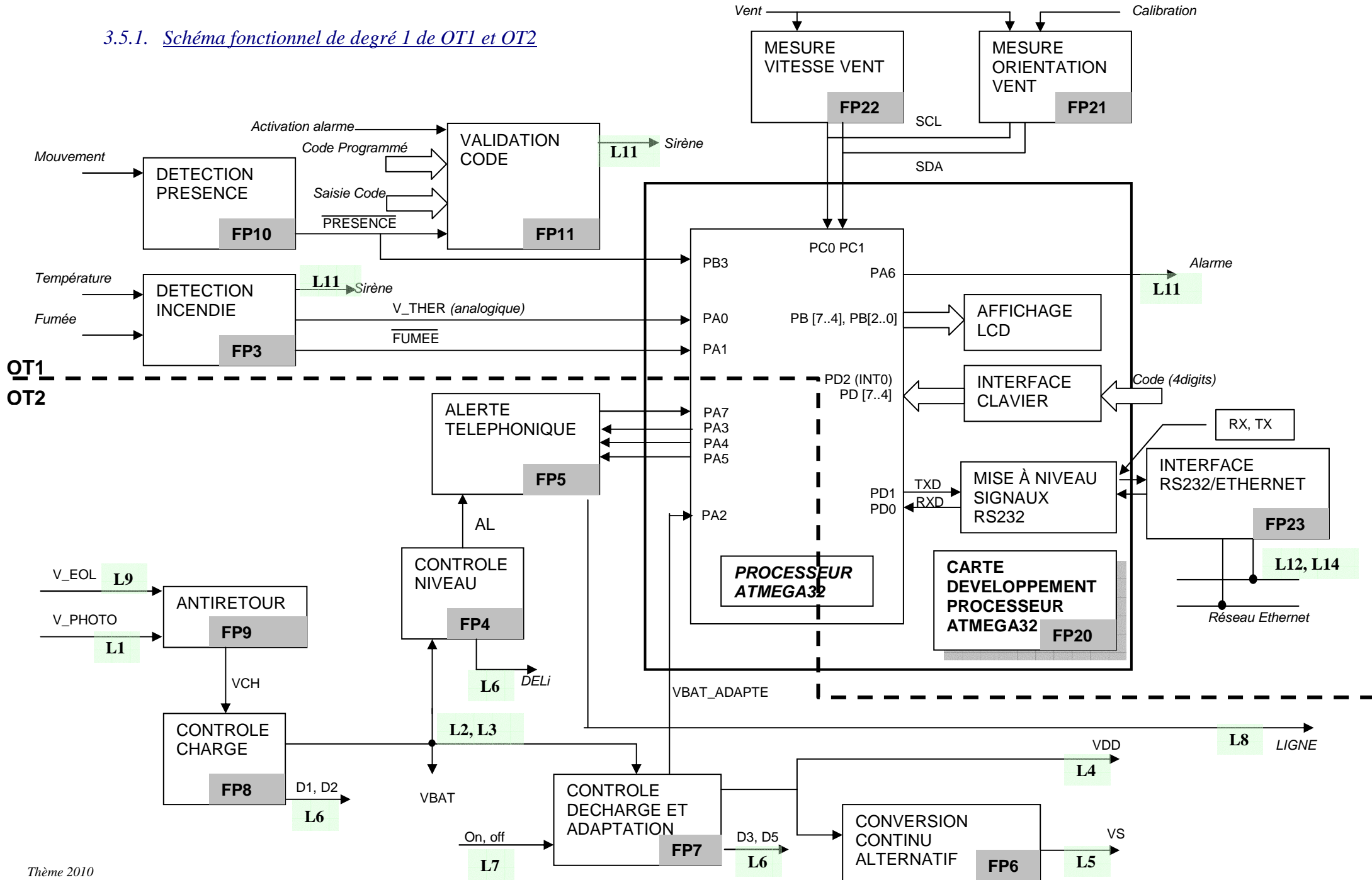
- gère la protection du domicile contre les risques d'incendie
- gère la protection contre les risques d'intrusions malveillantes
- mesure la vitesse et l'orientation du vent afin d'optimiser l'implantation de l'éolienne (à faire avant l'implantation définitive)

OT2: à partir d'énergie fournie sous forme de basses tensions (<18v), unipolaire, OT2 :

- gère la charge d'une batterie d'accumulateur en évitant les surcharges.
- gère la décharge de cette batterie en alimentant des récepteurs sous une tension continue de 12v et sous une tension sinusoïdale de 230v et en évitant les décharges excessives grâce à un délestage.
- fournit des informations visuelles sur l'état de cette batterie.
- déclenche une alarme (appel d'un numéro de téléphone) si la tension de la batterie devient inférieure à un seuil critique.

3.5. Etude fonctionnelle de degré 1

3.5.1. Schéma fonctionnel de degré 1 de OT1 et OT2



3.5.2. Définition des fonctions principales

Remarque : dans la suite les signaux type « SIGNAL » seront écrit sur la CAO PCAD « SIGNAL_ »

FP10 : Détection de présence.

Entrée(s):

- *Mouvement* : mouvement humain (ou autre...)

Sortie(s) :

- PRESENCE_ signal électrique binaire 0/5V résultant de la présence d'un mouvement.

FP11 : Validation code.

Entrée(s):

- PRESENCE_ signal électrique binaire 0/5V résultant de la présence d'un mouvement.
- *Activation alarme* : interrupteur manœuvré par l'utilisateur pour activer ou non la mise en alarme
- *Code programmé* : Programmation manuelle par switches des 4 codes préprogrammés
- *Saisie code* : appui sur une touche

Sortie(s) :

- *Sirène* : son issu du buzzer

FP20 : Gestion informatique (carte de développement ATMEGA).

Entrée(s):

- V_THER : PA0 signal analogique représentant la température ambiante
- FUMEE_ : PA1 signal binaire de durée 1s. Actif à 0 si présence de fumée
- VBAT_ADAPTE : PA2 Tension batterie
- CDCH : PA7 commande du chiffage
- PRESENCE_ : PB3 signal de présence humaine.
- *Code (4 digits)* PD2, PD[7..4] Dans la phase de programmation l'utilisateur entre le code d'accès secret. Dans la phase de fonctionnement normal, l'utilisateur doit taper ce code pour ne pas déclencher d'alarme intrusion
- PD1, PD0 : RX, TX liaison série asynchrone RS232 permettant d'envoyer le contenu de la page HTML

Sortie(s) :

- CE Sélection boîtier PA3
- CLK horloge PA4
- DATA donnée PA5
- *sirène*, signal d'alarme en cas d'incendie ou d'intrusion PA6
- SCL (sortie), SDA (entrée/sortie) PC0, PC1: bus série synchrone I2C. SCL est l'horloge, SDA les données
- *Afficheur LCD* PB[7 :0] : interface graphique

FP21 : Mesurer l'orientation du vent

Entrée(s):

- *Vent* : le vent permet d'orienter la girouette qui supporte une boussole
- *Calibration* : Action manuelle pour calibrer si besoin la boussole

Sortie(s) :

- SDA, SCL : signaux du bus série synchrone I2C (SDA données, SCL horloge) permettant de transmettre l'information d'orientation par rapport au pôle nord magnétique

FP22 : Mesure de la vitesse du vent.

Entrée(s):

- *Vent* : le vent fait tourner un disque percé de trous sur sa circonférence, à une fréquence proportionnelle à sa vitesse

Sortie(s) :

- SCL (sortie), SDA (entrée/sortie) : bus série synchrone I2C. SCL est l'horloge, SDA les données

FP23 : Convertir des données compatibles liaison RS232 en données compatibles réseau Ethernet avec gestion du protocole réseau Ethernet (et inversement).

Entrée(s):

- RX: signal de type RS232 (19200bits/s, caractère 8 bits, pas de parité) transmission sous forme série d'une suite de caractère ASCII représentant la page HTML demandée par le client

Sortie(s) :

- TX: signal de type RS232 (19200bits/s, caractère 8 bits, pas de parité) transmission sous forme série d'une suite de caractère ASCII représentant la requête (demande d'envoi de la page HTML) du client vers le serveur
- Réseau Ethernet : la connexion au réseau Ethernet permet la communication sur réseau local des informations entre client et serveur (voir ci-dessus la définition des signaux RX et TX)

FP3 : Détection d'incendie

Entrée(s):

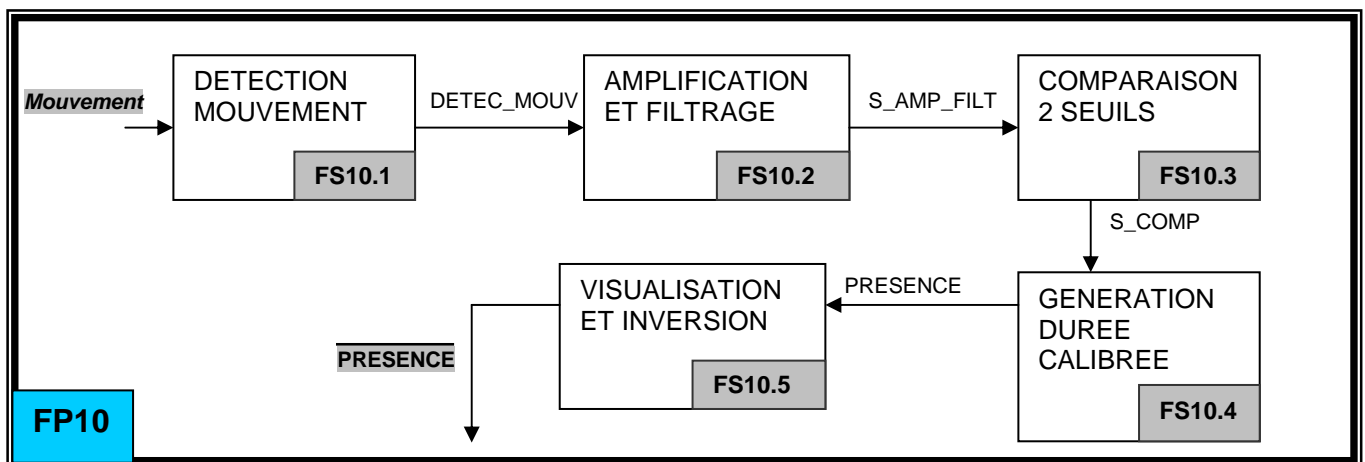
- *Température* : Température ambiante
- *Fumée* : fumée éventuelle due à un incendie

Sortie(s) :

- *Sirène* : son issu du buzzer
- *V_THER* : signal analogique représentant la température ambiante
- *FUMEE_* : signal binaire de durée 1s. Actif à 0 si présence de fumée

3.6. Etude fonctionnelle de degré 2

3.6.1. Schéma fonctionnel de degré 2 de FP10



3.6.2. Définition des fonctions secondaires de FP10

FS10.1 : Détecter un mouvement.

Entrée(s):

- *Mouvement*

Sortie(s) :

- *DETEC_MOUV*, signal électrique résultant de la présence d'un mouvement

FS10.2 : Amplifier et filtrer

Entrée(s):

- *DETEC_MOUV*, signal électrique résultant de la présence d'un mouvement

Sortie(s) :

- *S_AMP_FILT* : signal amplifié
 - Valeur moyenne : 2V
 - Amplitude maximale 5V (possibilité de saturation)

FS10.3 : Comparer un signal avec 2 seuils

Entrée(s):

- *S_AMP_FILT* : signal amplifié

Sortie(s) :

- *S_COMP* : Signal binaire
 - Si $VREF1 < S_AMP_FILT < VREF2$ alors $S_COMP = 1$
 - Si $S_AMP_FILT < VREF1$ ou $S_AMP_FILT > VREF2$ alors $S_COMP = 0$
- $VREF1 = 1.2V$, $VREF2 = 3.8V$ (environ).

FS10.4 : Générer une durée calibrée

Entrée(s):

- S_COMP : signal binaire

Sortie(s) :

- PRESENCE : signal actif à 1 de durée 1.5s lorsque S_COMP passe à 0

FS10.5 : Inverser un signal et visualiser

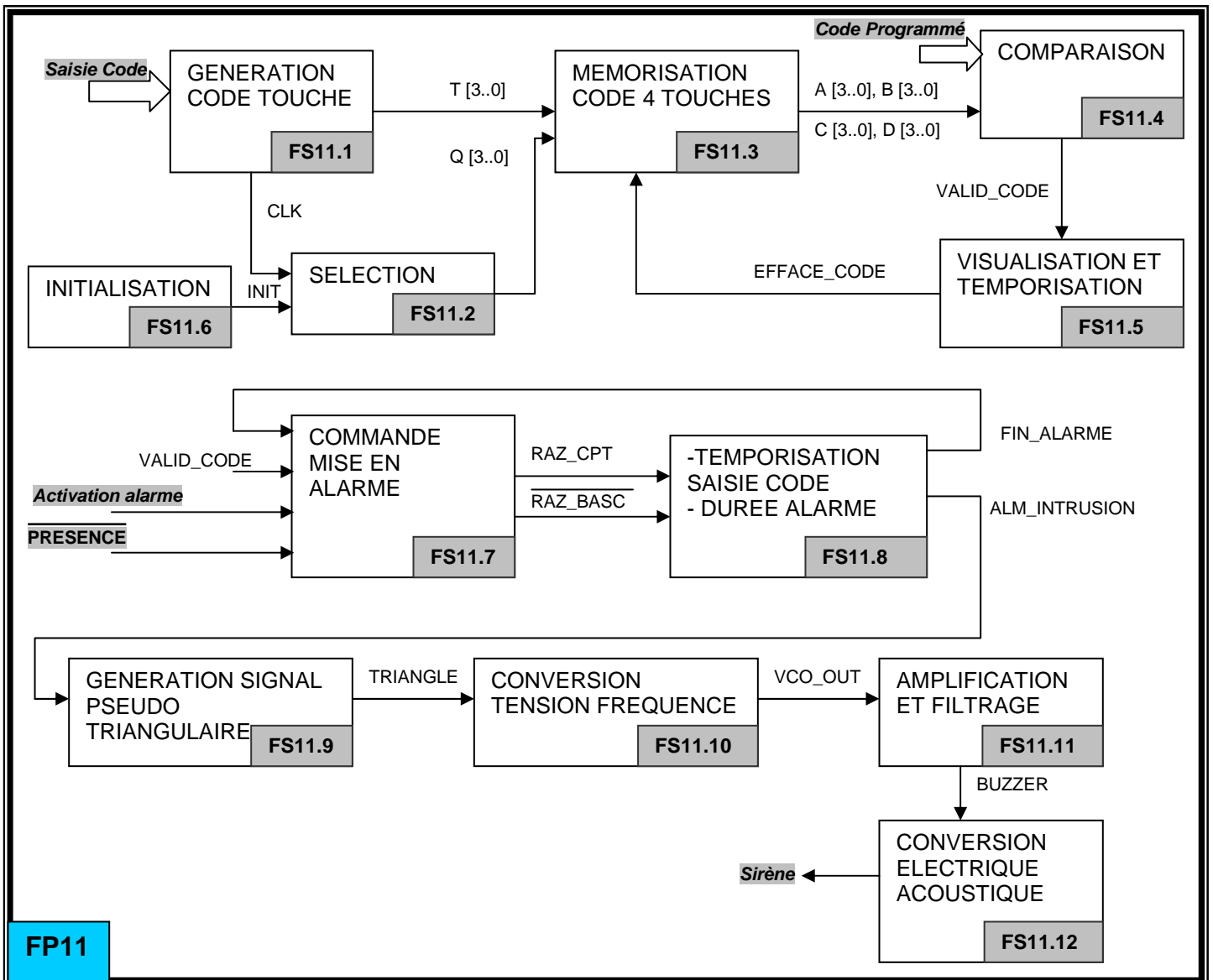
Entrée(s):

- PRESENCE : signal actif à 1 de durée 1.5s lorsque S_COMP passe à 0

Sortie(s) :

- PRESENCE_ : signal actif à 0 de durée 1.5s lorsque S_COMP passe à 0
- Information lumineuse sur la présence d'un mouvement dans le champ du capteur

3.6.3. Schéma fonctionnel de degré 2 de FP11



3.6.4. Définition des fonctions secondaires de FP11

FS11.1 : Générer le code la touche appuyée par l'utilisateur ainsi qu'un signal informant qu'un nouveau code est présent sur les sorties.

Entrée(s):

- Saisie code : appui sur une touche

Sortie(s) :

- T [3..0] code de la touche
- CLK signal informant de la présence d'un code sur les sorties

FS11.2 : Générer 4 signaux de commande de mémorisation des codes des 4 touches appuyées par l'utilisateur

Entrée(s):

- INIT remise à 0 de Q[3..0]
- CLK signal informant de la présence d'un code sur les sorties

Sortie(s):

- Q [3..0] Signaux de commande de mémorisation des codes des 4 touches appuyées par l'utilisateur.

FS11.3: Générer les codes des 4 touches appuyées successivement par l'utilisateur.

Entrée(s):

- T [3..0] code de la touche
- Q [3..0] signaux de commande de mémorisation des codes des 4 touches appuyées par l'utilisateur.
- EFFACE_CODE : lorsque 4 touches ont été appuyées, ce signal remet à 0 les sorties A[3..0], B[3..0], C[3..0], D[3..0]

Sortie(s) :

- A[3..0], B[3..0], C[3..0], D[3..0] 4 codes des 4 touches appuyées

FS11.4: Comparer les 4 codes des 4 touches appuyées aux 4 codes des 4 touches préprogrammées permettant l'accès.

Entrée(s):

- *Code programmé* : Programmation manuelle par switches des 4 codes préprogrammés
- A [3..0], B [3..0], C [3..0], D [3..0] 4 codes des 4 touches appuyées

Sortie(s) :

- VALID_CODE signal indiquant que le code des touches appuyées correspond au code préprogrammé

FS11.5: Indiquer à l'utilisateur que son code est valide par une impulsion lumineuse brève. Commande l'effacement des codes des touches appuyées par l'utilisateur pour une nouvelle utilisation.

Entrée(s):

- VALID_CODE signal indiquant que le code des touches appuyées correspond au code préprogrammé

Sortie(s) :

- EFFACE_CODE signal permettant l'effacement des 4 codes mémorisés
- Signal lumineux sur LED verte

FS11.6: Initialiser FS11.2 pour que la première touche appuyée soit comparée à la première touche programmée

Entrée(s):

- Tension d'alimentation 5V

Sortie(s) :

- INIT signal d'initialisation

FS11.7: Commander la mise en alarme

Entrée(s):

- VALID_CODE : signal indiquant que le code des touches appuyées correspond au code programmé.
- FIN_ALARME : signal permettant l'arrêt de l'alarme au bout d'un temps $T_{fin_alm} = 9s$
- PRESENCE_ : signal de présence humaine.
- *Activation alarme* : interrupteur manœuvré par l'utilisateur pour activer ou non la mise en alarme

Sortie(s) :

- RAZ_CPT : remise à 0 du compteur calibrant la durée nécessaire à l'utilisateur pour saisir le code d'accès
- RAZ_BASC : remise à 0 de la bascule de mémorisation ALARME

FS11.8: Générer 2 durées : durée nécessaire à l'utilisateur pour saisir le code d'accès, durée d'activation de la sirène

Entrée(s):

- RAZ_CPT : remise à 0 du compteur calibrant la durée nécessaire pour saisir le code d'accès
- RAZ_BASC : remise à 0 de la bascule de mémorisation ALARME

Sortie(s) :

- ALM_INTRUSION : signal d'activation de l'alarme au bout d'un temps $T_{alm} = 10s$ environ
- FIN_ALARME: signal permettant l'arrêt de l'alarme au bout d'un temps $T_{fin_alm} = 10s$ environ

FS11.9: Générer un signal « triangulaire », en fait portions de charge/décharge capacitive présent sur un temps très court par rapport à la constante de temps τ .

Entrée(s):

- ALM_INTRUSION : signal d'activation de l'alarme au bout d'un temps $T_{alm} = 10s$ environ

Sortie(s) :

- TRIANGLE : signal pseudo triangulaire de fréquence 0.8 Hz environ

FS11.10: Convertir tension/fréquence

Entrée(s):

- TRIANGLE, signal pseudo triangulaire de fréquence 0.8 Hz d'amplitude comprise entre $V_{MIN}=1.7V$ et $V_{MAX}=3.3V$

Sortie(s) :

- VCO_OUT : sortie de l'oscillateur contrôlé en tension
 - Si TRIANGLE = V_{MIN} VCO_OUT est de fréquence 315 Hz environ
 - Si TRIANGLE = V_{MAX} VCO_OUT est de fréquence 770 Hz environ

FS11.11: Amplifier le signal et élimine les fréquences harmoniques élevées

Entrée(s):

- VCO_OUT : sortie de l'oscillateur contrôlé en tension

Sortie(s) :

- BUZZER : signal analogique de commande du buzzer
 - Résultat du filtrage passe bas de VCO_OUT
 - Amplitude 4V
 - Valeur moyenne 2.5V

FS11.12 : Transformer un signal électrique en son

Entrée(s):

- BUZZER : signal analogique de commande du buzzer

Sortie(s) :

- Sirène : son issu du buzzer

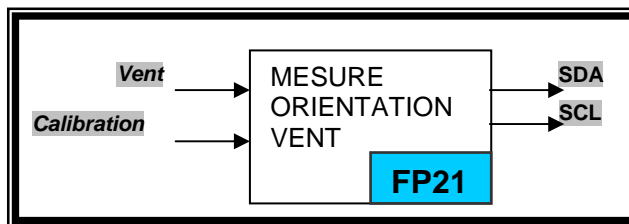
3.6.5. Schéma fonctionnel de degré 2 de FP20

Voir schéma fonctionnel de degré 1

3.6.6. Définition des fonctions secondaires de FP20

Fonctions logicielles programmation C sur ATMEGA32. Voir la partie « FP20 » des travaux élèves.

3.6.7. Schéma fonctionnel de degré 2 de FP21



3.6.8. Définition des fonctions secondaires de FP21

FP21 : Mesurer l'orientation du vent

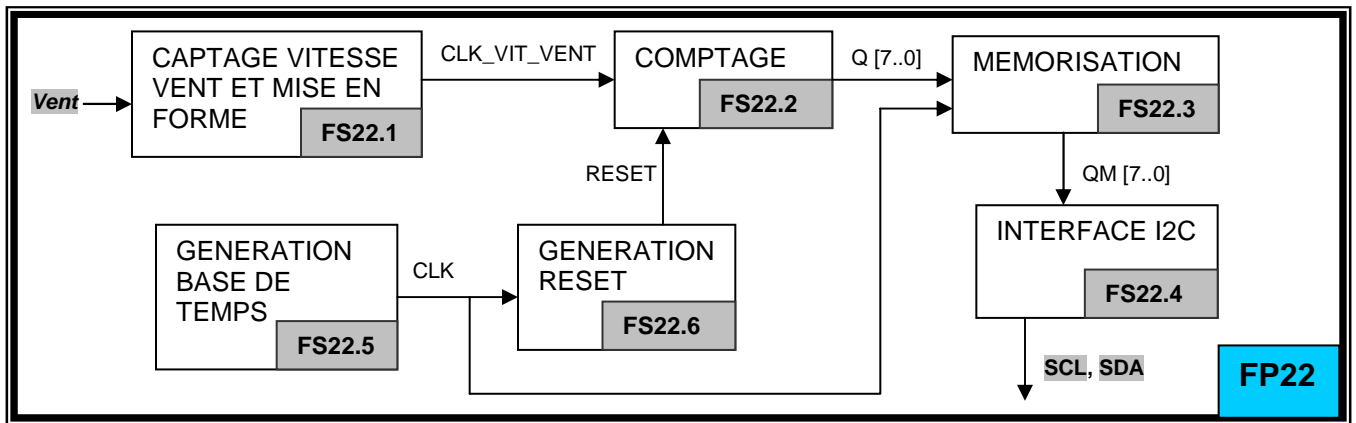
Entrée(s):

- Vent : le vent permet d'orienter la girouette qui supporte une boussole
- Calibration : Action manuelle pour calibrer si besoin la boussole

Sortie(s) :

- SDA, SCL : signaux du bus série synchrone I2C (SDA données, SCL horloge) permettant de transmettre l'information d'orientation par rapport au pôle nord magnétique

3.6.9. Schéma fonctionnel de degré 2 de FP22



3.6.10. Définition des fonctions secondaires de FP22

FS22.1 : Capturer la vitesse du vent et mettre en forme le signal

Entrée(s) :

- *Vent* : un disque percé de trous tourne à une vitesse angulaire proportionnelle à la vitesse du vent

Sortie(s) :

- *CLK_VIT_VENT* : signal binaire de fréquence multiple de la vitesse angulaire du disque

FS22.2 : Compter des impulsions

Entrée(s) :

- *CLK_VIT_VENT* : signal binaire de fréquence multiple de la vitesse angulaire du disque
- *RESET* : signal de remise à 0 du compteur actif à 1

Sortie(s) :

- *Q [7..0]* : Nombre d'impulsions du signal *CLK_VIT_VENT* entre 2 *RESET*

FS22.3 : Mémoriser

Entrée(s) :

- *Q [7..0]* : Nombre d'impulsions du signal *CLK_VIT_VENT* entre 2 *RESET* successifs
- *CLK* : signal binaire d'ordre de mémorisation

Sortie(s) :

- *QM [7..0]* : Nombre d'impulsions du signal *CLK_VIT_VENT* entre 2 *RESET* mémorisé

FS22.4 : Interfacer avec le bus I2C

Entrée(s) :

- *QM [7..0]* : Nombre d'impulsions du signal *CLK_VIT_VENT* entre 2 *RESET* mémorisé

Sortie(s) :

- *SDA, SCL* : signaux du bus série synchrone I2C (*SDA* données, *SCL* horloge) permettant de transmettre le nombre d'impulsions reçues entre 2 *RESET* successifs

FS22.5 : Générer une base de temps

Sortie(s) :

- *CLK* : signal binaire $T_H = 1s, T_L = 9ms$

FS22.6 : Générer un signal de RESET (remise à 0)

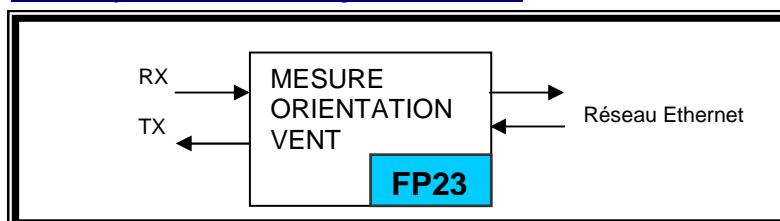
Entrée(s) :

- *CLK* : signal binaire $T_H = 1s, T_L = 9ms$

Sortie(s) :

- *RESET* : signal de remise à 0 du compteur actif à 1. *RESET* doit être retardé par rapport à *CLK* d'une durée supérieure à $15\mu s$ (set up time du composant de mémorisation). $100\mu s$ minimum imposé

3.6.11. Schéma fonctionnel de degré 2 de FP23



3.6.12. Définition des fonctions secondaires de FP23

FP23 : Convertir des données compatibles liaison RS232 en données compatibles réseau Ethernet avec gestion du protocole réseau Ethernet (et inversement).

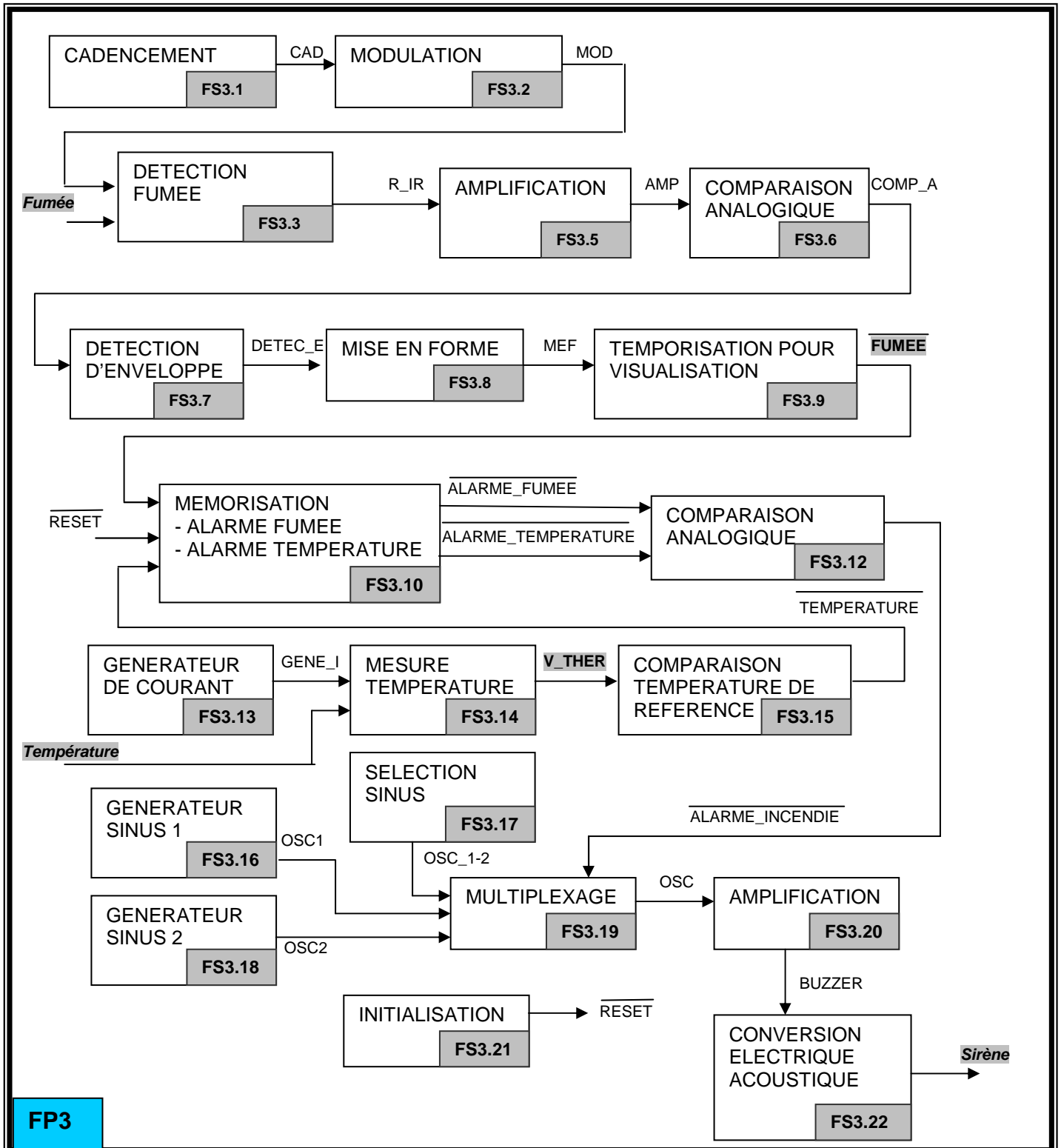
Entrée(s):

- RX: signal de type RS232 (19200bits/s, caractère 8 bits, pas de parité) transmission sous forme série d'une suite de caractère ASCII représentant la page HTML demandée par le client

Sortie(s) :

- TX: signal de type RS232 (19200bits/s, caractère 8 bits, pas de parité) transmission sous forme série d'une suite de caractère ASCII représentant la requête (demande d'envoi de la page HTML) du client vers le serveur
- Réseau Ethernet : la connexion au réseau Ethernet permet la communication sur réseau local des informations entre client et serveur (voir ci-dessus la définition des signaux RX et TX)

3.6.13. Schéma fonctionnel de degré 2 de FP3



FP3

3.6.14. Définition des fonctions secondaires de FP3

FS3.1: Cadencer les émissions de la diode infra rouge

Sortie(s) :

- CAD signal de fréquence 1.2Hz environ et rapport cyclique 1/500 environ

FS3.2: Moduler le signal en largeur d'impulsion

Entrée(s):

- CAD : signal de fréquence 1.2Hz environ et rapport cyclique 1/500 environ

Sortie(s) :

- MOD : signal de type train d'impulsions :
 - Si CAD = 0 ; MOD = 0
 - Si CAD = 1 ; MOD est un signal de fréquence 5 kHz

FS3.3 : Détecter une présence de fumée

Entrée(s):

- MOD : signal de type train d'impulsions
- Fumée

Sortie(s) :

- R_IR: Signal électrique représentant le signal infra rouge

FS3.4: Non implantée

FS3.5: Amplifier

Entrée(s):

- R_IR: Signal électrique représentant le signal infra rouge

Sortie(s) :

- AMP : Signal R_IR amplifié :
 - Elimination de la composante continue de R_IR
 - Décalage (offset) de 2.5V
 - Amplification de 500

FS3.6: Comparer 2 signaux analogiques

Entrée(s):

- AMP : Signal R_IR amplifié

Sortie(s) :

- COMP_A: Signal électrique binaire (0/5V)
 - Si AMP est supérieur à une tension continue réglable VREF1 alors COMP_A = 0V
 - Sinon COMP_A est un signal binaire (0/5V) de même fréquence que AMP

FS3.7: Détecter l'enveloppe d'un signal

Entrée(s):

- COMP_A: Signal électrique binaire (0/5V)

Sortie(s) :

- DETEC_E: Enveloppe du signal COMP_A

FS3.8: Mettre en forme

Entrée(s):

- DETEC_E: Enveloppe du signal COMP_A

Sortie(s) :

- MEF : signal binaire dépourvu d'oscillations

FS3.9: Temporiser pour visualiser

Entrée(s):

- MEF : signal binaire dépourvu d'oscillations

Sortie(s) :

- FUMEE_ : signal binaire de durée 1s.

FS3.10: Mémoriser les informations de température et de présence fumée

Entrée(s):

- FUMEE_ : signal binaire de durée 1s.
- TEMPERATURE : signal binaire. Actif à 0 si présence de température supérieure à un seuil critique
- RESET_ : signal de remise à 1 de FUMEE_ et TEMPERATURE

Sortie(s) :

- ALARME_FUMEE_ : signal binaire, mémorisation de FUMEE_
- ALARME_TEMPERATURE_ : signal binaire mémorisation de TEMPERATURE

FS3.11 : non implantée

FS3.12: Comparer les informations de température et de présence fumée par rapport à une tension fixe VREF2. Visualiser le type d'alarme (Fumée ou température ou les 2).

Entrée(s):

- ALARME_FUMEE_ : signal binaire, mémorisation de FUMEE_
- ALARME_TEMPERATURE_ : signal binaire mémorisation de TEMPERATURE

Sortie(s) :

- ALARME_INCENDIE_ : signal binaire actif à 0 si présence de fumée ou de température excessive

FS3.13: Générer un courant**Sortie(s) :**

- GENE_I : signal de tension variable et de courant fixe égal à 1mA

FS3.14: Mesurer la température**Entrée(s) :**

- GENE_I : signal de tension variable et de courant fixe égal à 1mA
- Température : grandeur physique

Sortie(s) :

- V_THER : Tension fonction de la température mesurée

FS3.15: Comparer la tension fonction de la température mesurée à la tension VREF3 représentant la température critique**Entrée(s) :**

- V_THER : Tension fonction de la température mesurée

Sortie(s) :

- TEMPERATURE : Signal binaire
 - Si V_THER est inférieure à VREF3, TEMPERATURE = 0
 - Si V_THER est supérieure à VREF3, TEMPERATURE = 1

FS3.16: Générer un signal sinusoïdal**Sortie(s) :**

- OSC1 : signal sinusoïdal de fréquence 1.6kHz, d'amplitude 4 V c-c

FS3.17: Sélectionner un des 2 signaux sinusoïdaux**Sortie(s) :**

- OSC_1-2 : signal carré de période 1.2s rapport cyclique 1/12

FS3.18: Générer un signal sinusoïdal**Sortie(s) :**

- OSC2 : signal sinusoïdal de fréquence 770 Hz, d'amplitude 4 V c-c

FS3.19: Multiplexer**Entrée(s) :**

- OSC1 : signal sinusoïdal de fréquence 1.6kHz, d'amplitude 4 V c-c
- OSC2 : OSC2 : signal sinusoïdal de fréquence 770 Hz, d'amplitude 4 V c-c
- OSC_1-2 : signal carré de période 1.2s rapport cyclique 1/12
- ALARME_INCENDIE_ : signal binaire actif à 0 si présence de fumée ou de température excessive

Sortie(s) :

- OSC : signal sinusoïdal égal soit à OSC1 soit à OSC2

FS3.20: Amplifier un signal sinusoïdal**Entrée(s) :**

- OSC : signal sinusoïdal de fréquence 1.6kHz ou 770 Hz, d'amplitude 4 V

Sortie(s) :

- BUZZER : signal sinusoïdal de fréquence 1.6kHz ou 770 Hz, d'amplitude maximale comprise entre -5V et +5V

FS3.21: Initialiser et ré initialiser le système

Entrée(s) :

- Action manuelle sur un bouton poussoir

Sortie(s) :

- RESET_ : signal d'initialisation du système actif à 0. Lors de la mise sous tension, RESET doit rester à 0 pendant 10 ms environ.

FS3.22: Convertir un signal électrique en un signal sonore

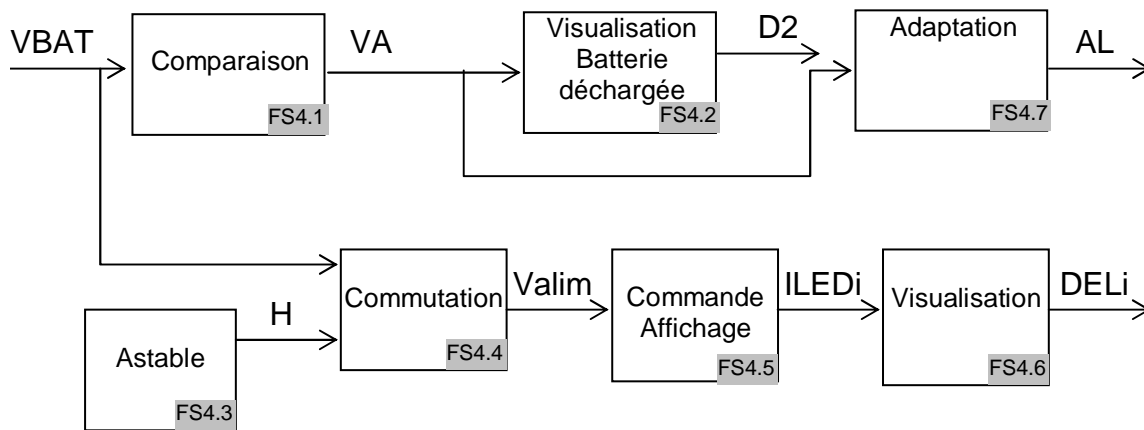
Entrée(s) :

- BUZZER : signal sinusoïdal de fréquence 1.6kHz ou 770 Hz, d'amplitude maximale comprise entre -5V et +5V

Sortie(s) :

- Sirène, signal sonore de fréquence 1600 Hz ou 770 Hz

3.6.15. Schéma fonctionnel de degré 2 de FP4



3.6.16. Définition des fonctions secondaires de FP4

FS4.1: Comparaison

Entrée(s):

- VBAT: tension aux bornes de la batterie ($8v < VBAT < 15v$)

Sortie(s) :

- VA=VBAT si $VBAT < 9v$ sinon VA=0v

FS4.2: Visualisation: batterie déchargée

Entrée(s):

- VA :

Sortie(s) :

- D2 : information lumineuse, DEL allumée si VA=VBAT

FS4.3: Astable

Sortie(s) :

- H : information logique (0v-VBAT), signal rectangulaire de fréquence 20Hz et de rapport cyclique 0,1

FS4.4: Commutation d'alimentation

Entrée(s):

- H

Sortie(s) :

- Valim: tension d'alimentation de FS4.5 , Valim = VBAT si H est au niveau logique haut sinon Valim= 0v

FS4.5: Commande affichage

Entrée(s):

- VBAT

Sortie(s) :

- ILED1 à ILED10: sources de courant commandées

VBAT	ILED1	ILED2	ILED3	ILED4	ILED5	ILED6	ILED7	ILED8	ILED9	ILED10
<9v	0	0	0	0	0	0	0	0	0	0
9v à 9,5v	Icc	0	0	0	0	0	0	0	0	0
9,5v à 10v	Icc	Icc	0	0	0	0	0	0	0	0
...										
13v à 13,5v	Icc	Icc	Icc	Icc	Icc	Icc	Icc	Icc	Icc	0
13,5v à 14v	Icc	Icc	Icc	Icc	Icc	Icc	Icc	Icc	Icc	Icc

FS4.6: Visualisation: tension batterie

Entrée(s):

- ILED1 à ILED10 :

Sortie(s) :

- DEL1 à DEL10 : information lumineuse, DELi allumée si ILEDi= Icc.

FS4.7: Adaptation

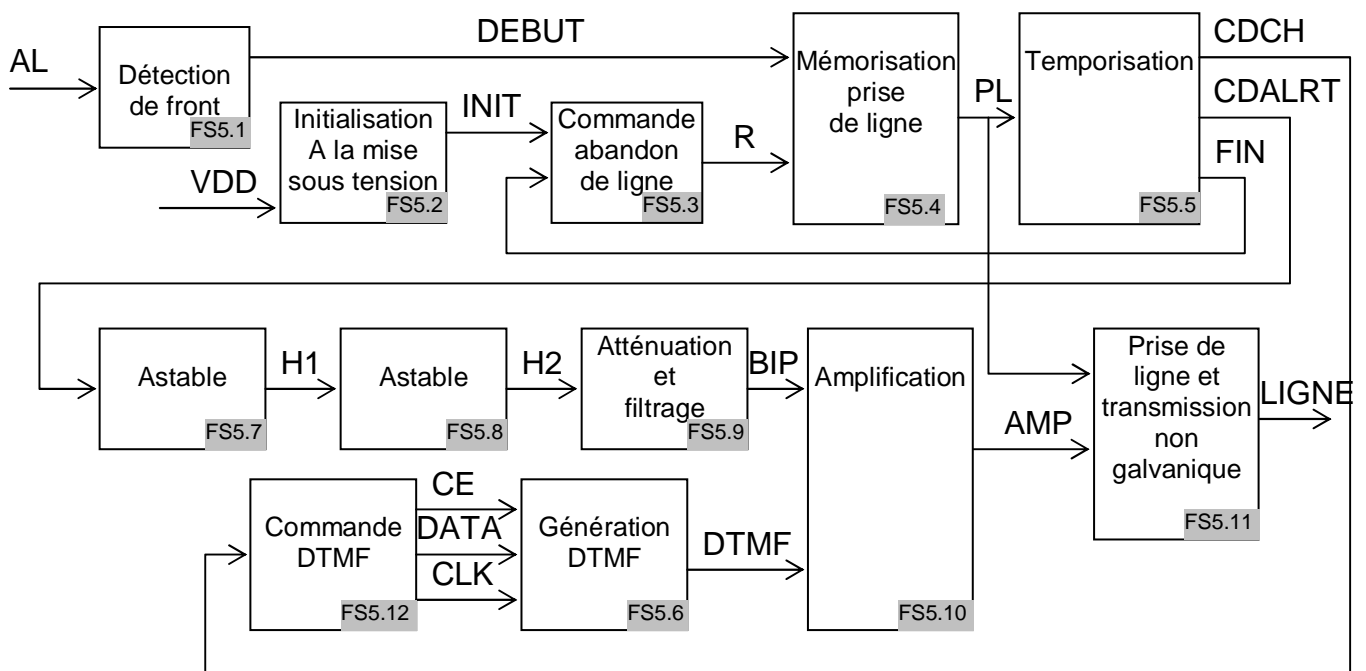
Entrée(s):

- VA :

Sortie(s) :

- AL: information logique : état d'un transistor à collecteur ouvert transistor saturé si VA= VBAT , bloqué sinon.

3.6.17. Schéma fonctionnel de degré 2 de FP5



3.6.18. Définition des fonctions secondaires de FP5

FS5.1: Détection de fronts descendants

Entrée(s):

- AL

Sortie(s) :

- DEBUT: un front descendant sur AL génère une impulsion positive sur DEBUT

FS5.2: Initialisation à la mise sous tension

Entrée(s):

- VDD: tension 12v fournie par la batterie.

Sortie(s) :

- INIT: tension correspondant à un niveau logique haut à la mise sous tension de FP5.

FS5.3: Génération de la commande d'abandon de la ligne téléphonique

Entrée(s):

- FIN et INIT:

Sortie(s) :

- R: information logique

FS5.4: Mémorisation de la prise de ligne

Entrée(s):

- DEBUT et R: informations logiques

Sortie(s) :

- PL: information logique commandant la prise de ligne

FS5.5: Temporisation

Entrée(s):

- PL

Sortie(s) :

- CDCH, CDALRT et FIN: informations logiques actives au niveau haut. Si PL est inactive les trois sorties sont inactives sinon, les sorties deviennent actives après de durées TCDCH, TCDALRT et TFIN

FS5.6: Génération de signaux DTMF (dual-tone multi-frequency)

Entrée(s):

- CE, CLK et DATA: informations logiques assurant une liaison série entre FP20 et FP5

Sortie(s) :

- DTMF: signal DTMF

FS5.7: Astable commandé

Entrée(s):

- CDALRT

Sortie(s) :

- H1: information logique, signal rectangulaire de période 0,5s de rapport cyclique égal à 0,5 lorsque CDALRT est au niveau logique haut. Sinon H1 est au niveau logique bas.

FS5.8: Astable commandé

Entrée(s):

- H1

Sortie(s) :

- H2: information logique, signal rectangulaire de fréquence 1khz de rapport cyclique égal à 0,5 lorsque H1 est au niveau logique haut. Sinon niveau logique bas.

•

FS5.9: Atténuation et filtrage

Entrée(s):

- H2

Sortie(s) :

- BIP: signal H2 débarrassé de sa composante continue et atténué.

FS5.10: Amplification

Entrée(s):

- BIP et DTMF

Sortie(s) :

- AMP.

FS5.11: Prise de ligne et transmission non galvanique

Entrée(s):

- CAMP et PL

Sortie(s) :

- LIGNE: ligne téléphonique. Si PL est à l'état haut, la ligne téléphonique est prise (circuit fermé sur une résistance en série avec une bobine) et le signal AMP est transmis à la ligne. sinon la ligne téléphonique est en circuit ouvert.

FS5.12: Commande DTMF (réalisée à l'aide d'une carte ATMEGA32)

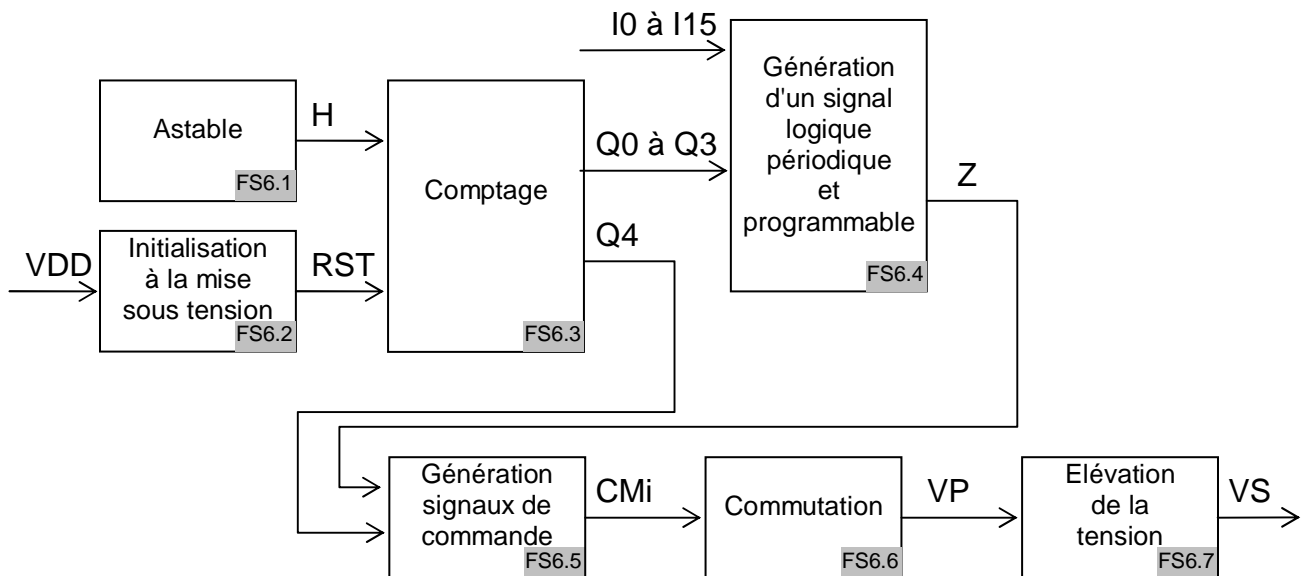
Entrée(s):

- CDCH

Sortie(s) :

- CE, DATA et CLK

3.6.19. Schéma fonctionnel de degré 2 de FP6



3.6.20. Définition des fonctions secondaires de FP6

FS6.1: Astable

Sortie(s) :

- H: signal rectangulaire de fréquence 1,6khz.

FS6.2: Initialisation à la mise sous tension

Entrée(s):

- VDD: tension 12v fournie FP7.

Sortie(s) :

- RST: impulsion positive lorsque VDD passe de 0v à 12v

FS6.3: Comptage

Entrée(s):

- RST et H

Sortie(s) :

- Q4 à Q0: Information numérique sur 5 bits

FS6.4: Génération d'un signal logique périodique et programmable

Entrée(s):

- Q3 à Q0
- IO à I15: informations manuelles (état de switches)

Sortie(s) :

- Z: signal logique

FS6.5: Génération des signaux de commande

Entrée(s):

- Z et Q4

Sortie(s) :

- CM1 et CM2, informations logiques: si Q4 est au niveau logique haut, CM2 = Z et CM1= "niveau bas" sinon CM2 = "niveau bas" et CM1= Z

FS6.6: Commutation**Entrée(s):**

- CM1 et CM2

Sortie(s) :

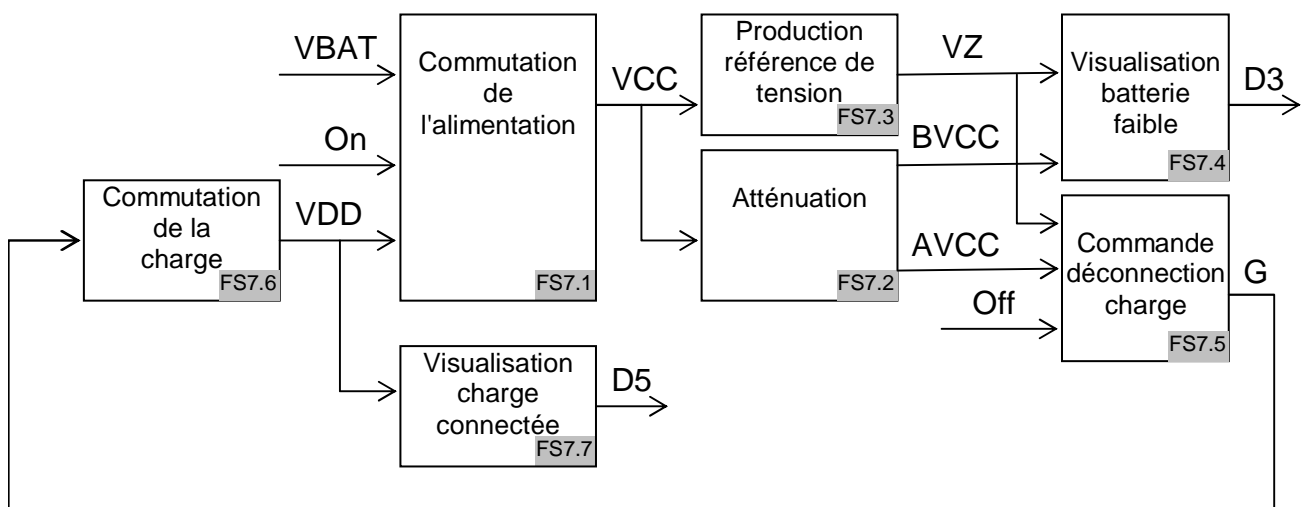
- VP, tension appliquée au primaire d'un transformateur.

FS6.7: Elévation de la tension**Entrée(s):**

- VP

Sortie(s) :

- VS: tension au secondaire du transformateur.

3.6.21. Schéma fonctionnel de degré 2 de FP73.6.22. Définition des fonctions secondaires de FP7

FS7.1: Commutation de l'alimentation: afin d'économiser l'énergie de la batterie, FP7 n'est plus alimentée lorsque la batterie est déconnectée de la charge

Entrée(s):

- VDS d'un transistor MOS (bloqué ou saturé).
- On: Information manuelle, bouton poussoir, contact normalement ouvert
- VBAT ; tension batterie

Sortie(s) :

- VCC: tension égale à VBAT si VDS=0v (MOS passant) ou si le contact "On" est fermé sinon VCC=0v.

FS7.2: Atténuation**Entrée(s):**

- VCC

Sortie(s) :

- AVCC= VCC * a (avec a<1)
- BVCC= VCC * b (avec b<1)

FS7.3: Production d'une référence de tension**Entrée(s):**

- VCC

Sortie(s) :

- VZ: tension constante lorsque VCC=VBAT

FS7.4: Visualisation: batterie faible

Entrée(s):

- VZ et BVCC

Sortie(s) :

- Information lumineuse: D3 (DEL rouge)

FS7.5: Commande de la déconnection de la charge

Entrée(s):

- Tensions VZ , AVCC
- Off: Information manuelle, bouton poussoir, contact normalement ouvert

Sortie(s) :

- G: information logique, G=VCC ou G=0v

FS7.6: Commutation de la charge

Entrée(s):

- G

Sortie(s) :

- Tension VDD=VBAT si G=VCC et VDD = 0v si G= 0v

FS7.7: Visualisation: charge connectée

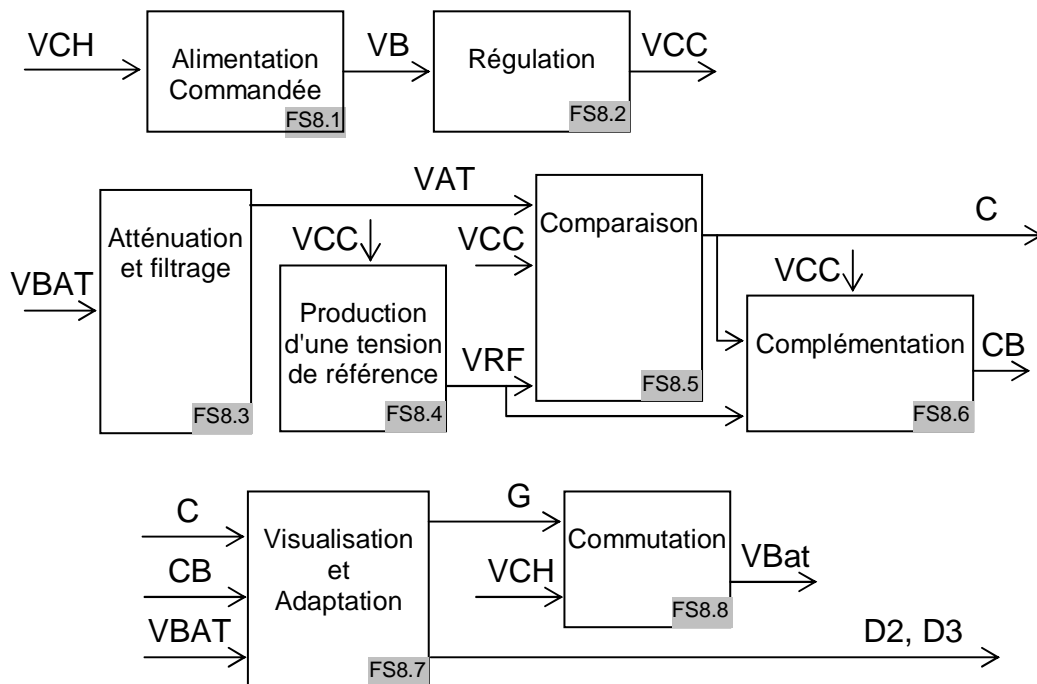
Entrée(s):

- VDD

Sortie(s) :

- Information lumineuse: D5 (DEL verte)

3.6.23. Schéma fonctionnel de degré 2 de FP8



3.6.24. Définition des fonctions secondaires de FP8

FS8.1: Alimentation commutée

Entrée(s):

- VCH : tension produite par l'éolienne ou les cellules photovoltaïques ($VCH < 18v$)

Sortie(s) :

- VB : $VB = VCH$ si $VCH > VCH_{mini}$ sinon $VB = 0v$

FS8.2: Régulation**Entrée(s):**

- VB

Sortie(s) :

- VB: $V_B = V_{CH}$ $V_{CH} > V_{CH_{\min}}$ sinon $V_B = 0v$

FS8.3: Atténuation et filtrage**Entrée(s):**

- VBAT

Sortie(s) :

- $V_{AT} = k V_{BAT}$ ($k < 1$)

FS8.4: Production d'une référence de tension**Sortie(s) :**

- $V_{RF} = V_{CC}/2$

FS8.5: Comparaison à deux seuils**Entrée(s):**

- V_{AT} et V_{RF}

Sortie(s) :

- C: information logique

FS8.6: Complémentation**Entrée(s):**

- V_{RF} et C

Sortie(s) :

- $CB = \overline{C}$

FS8.7: Visualisation et Adaptation**Entrée(s):**

- C, CB et V_{BAT}

Sortie(s) :

- information visuelle (DEL D2 et D3)
- G: information logique $G = V_{BAT}$ ou $G = 0v$

FS8.8: Commutation**Entrée(s):**

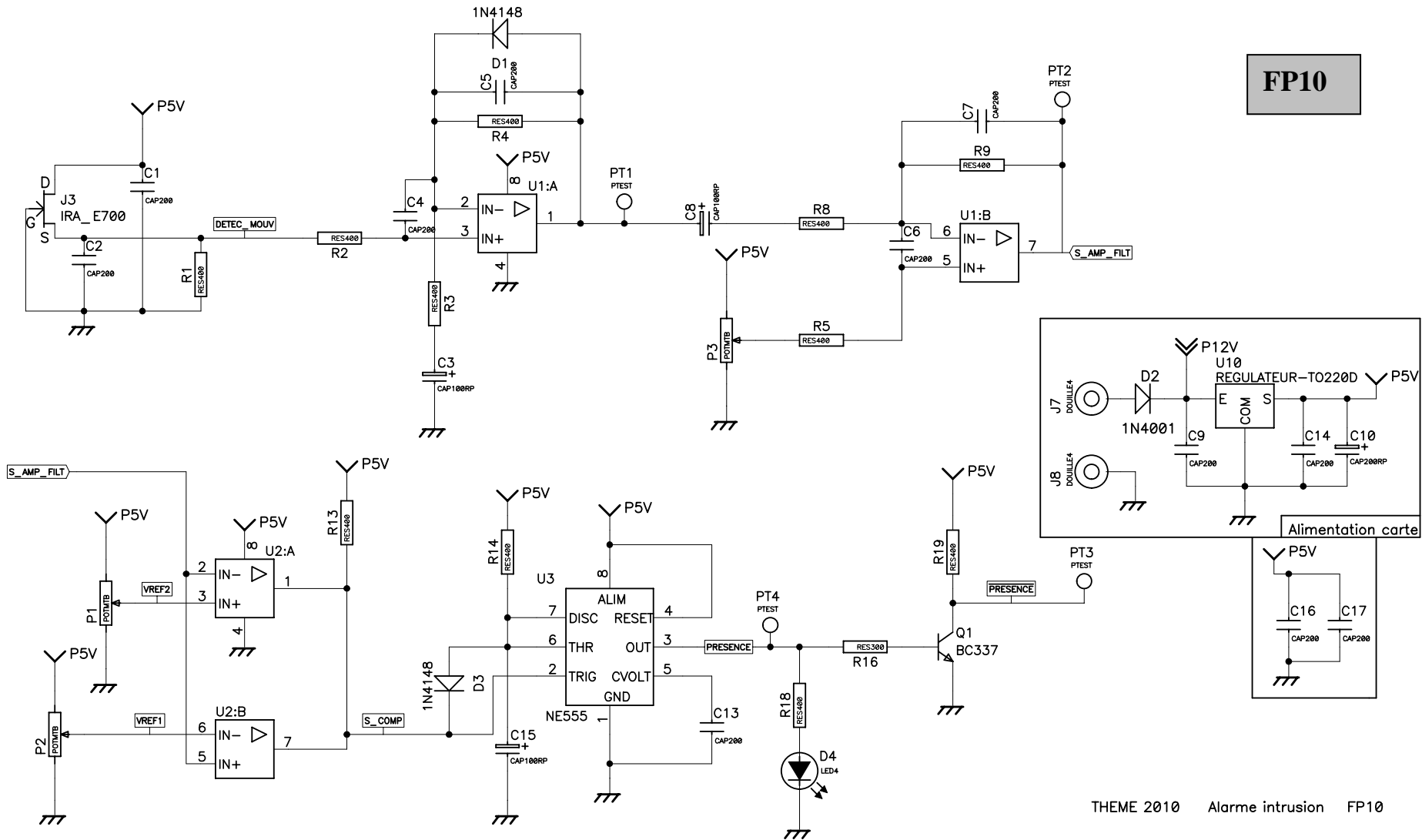
- G et V_{CH}

Sortie(s) :

- si $G = V_{BAT}$: la batterie est reliée à V_{CH} (charge) sinon, la batterie est déconnectée de V_{CH}

4. Schémas structurels

4.1. FP10



FP10

THEME 2010 Alarme intrusion FP10

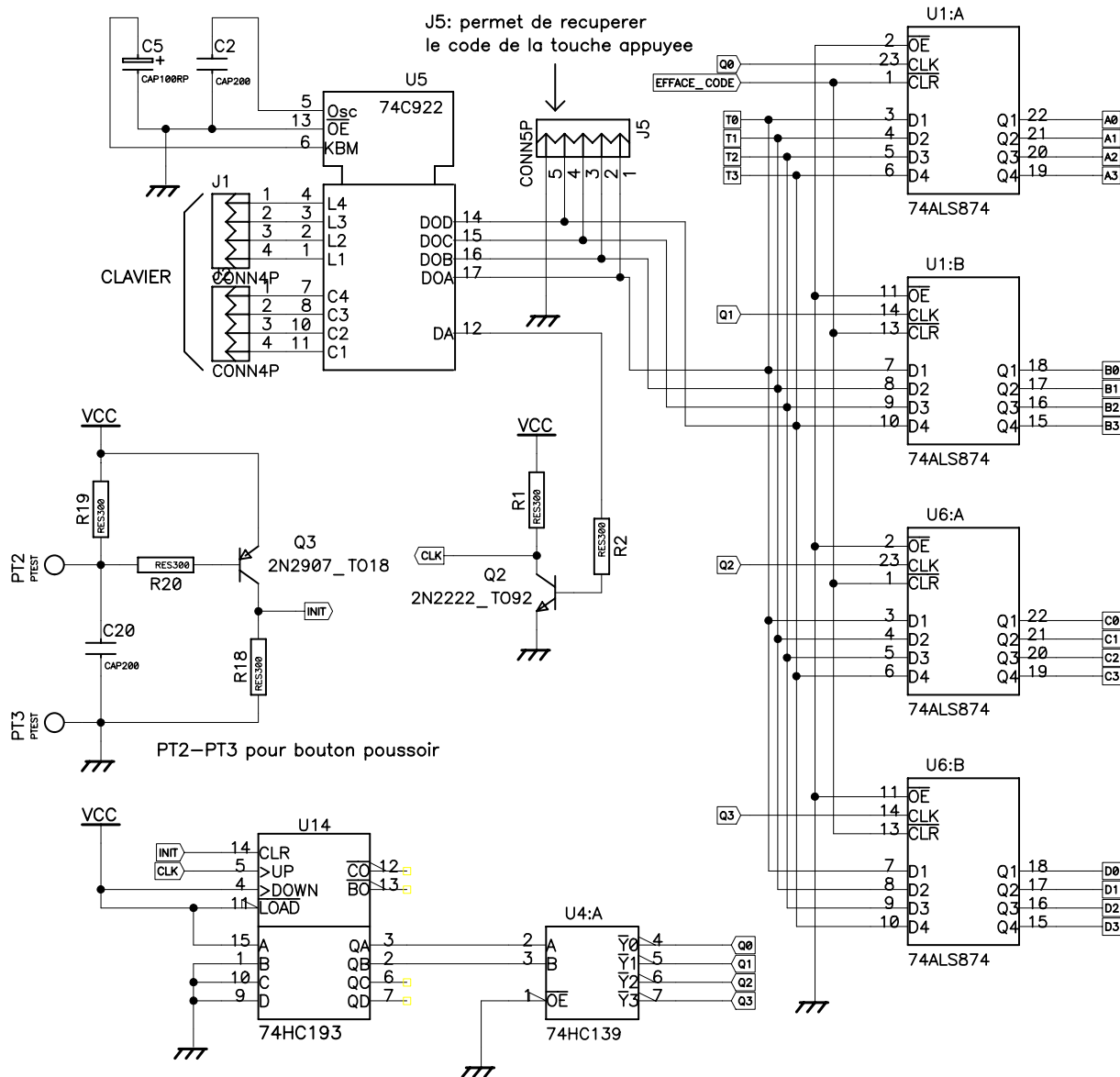
detection presence

2

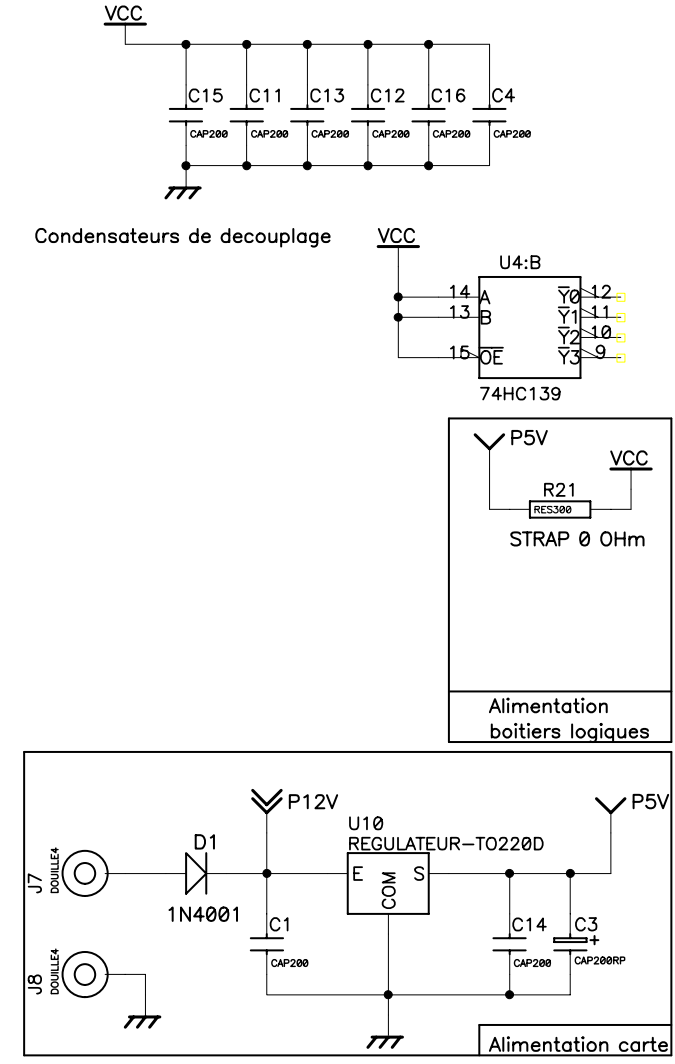
Detection IR.sch

Lycee Cournot
1 1

4.2. FP11



FP11 1/3

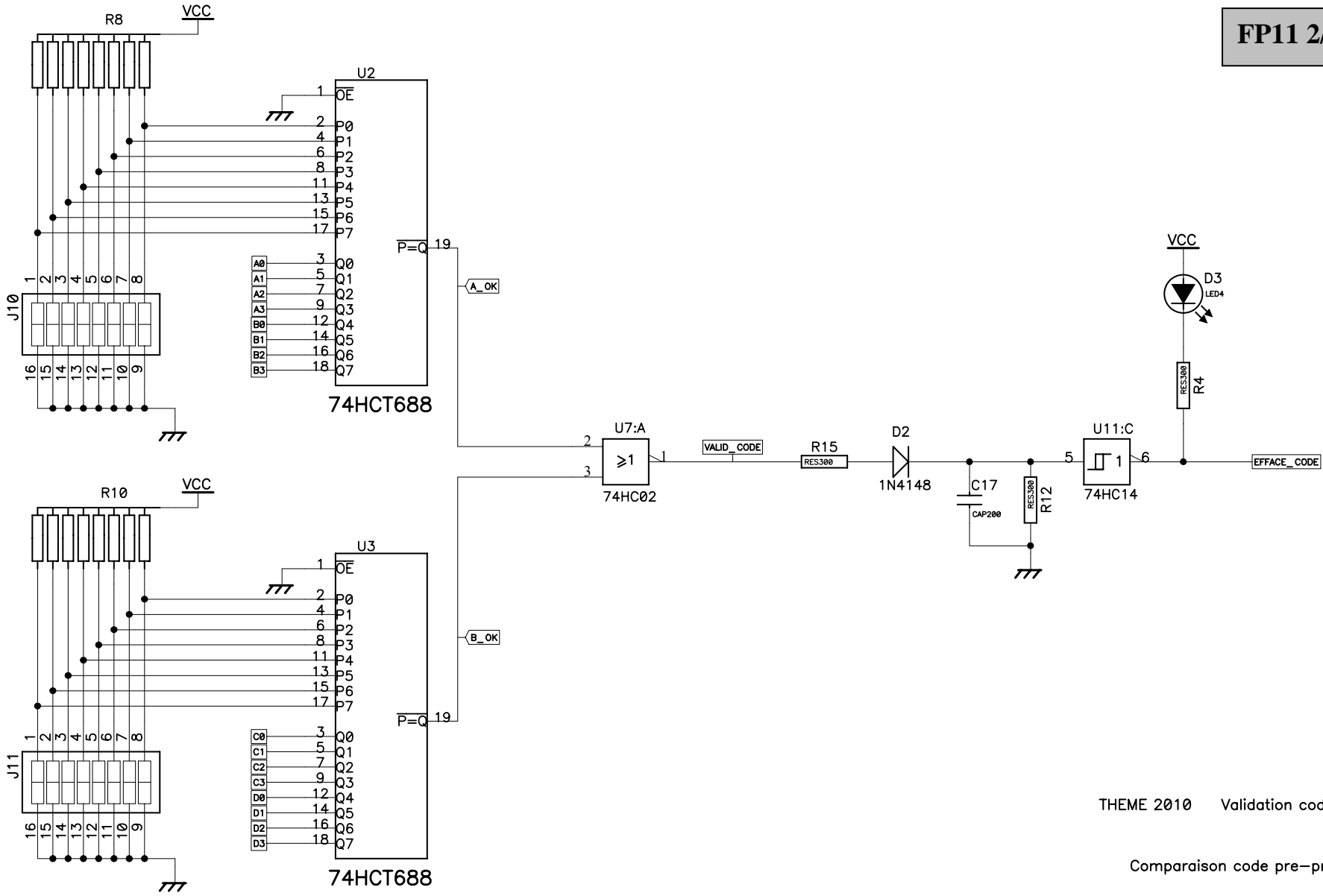


THEME 2010 Validation code FP11

Saisie et memorisation code 3

Alarme_intrusion_tout.sch 1 3
Lycee Cournot

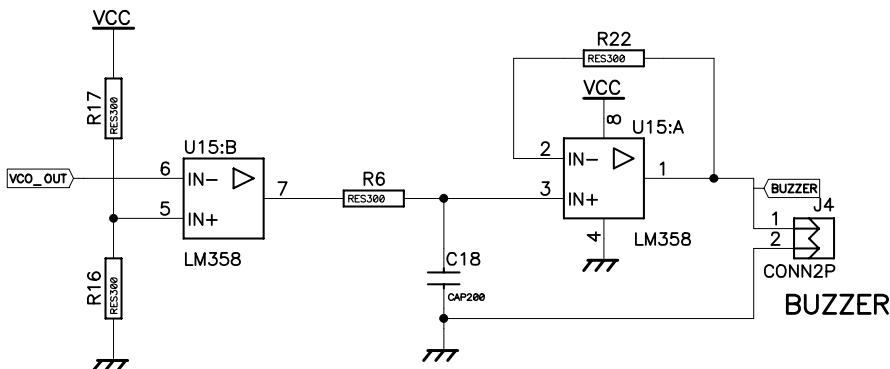
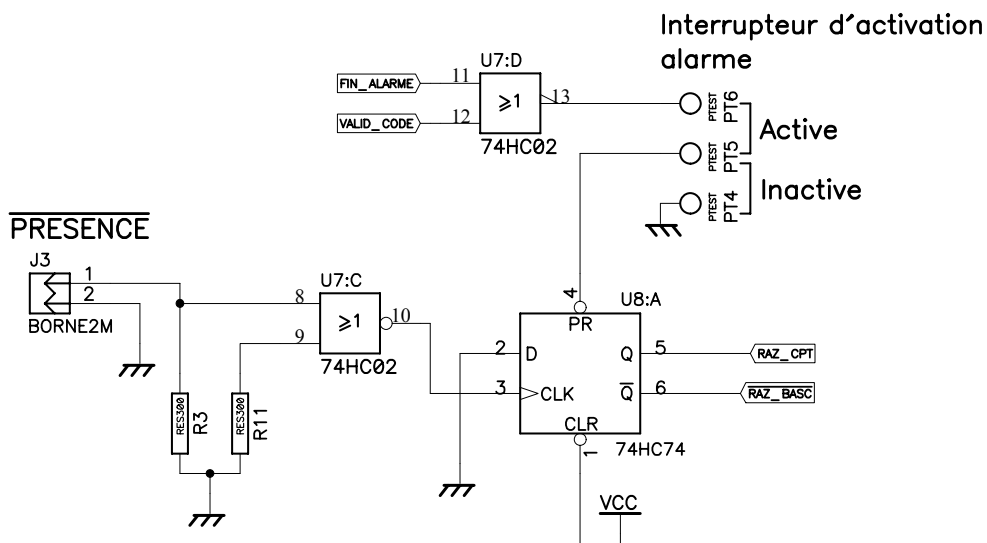
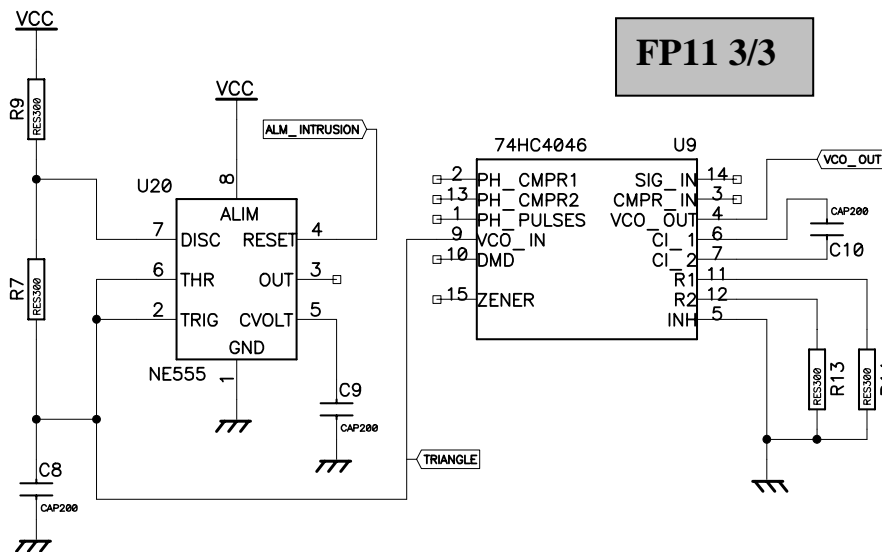
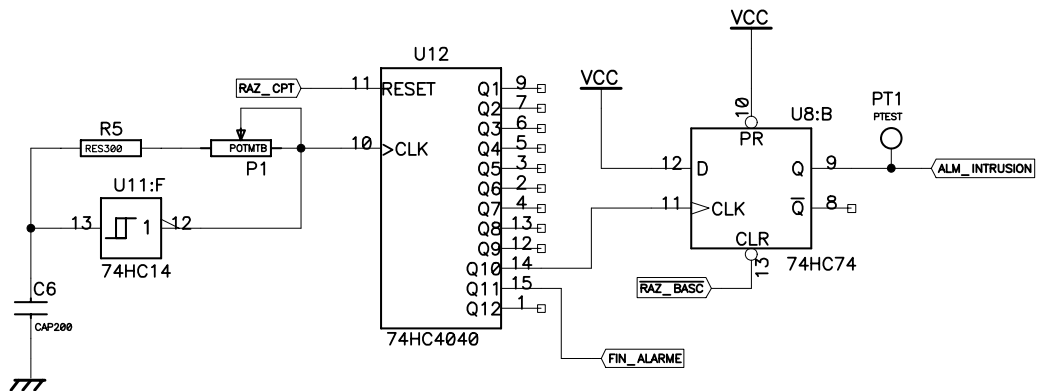
FP11 2/3



THEME 2010 Validation code FP11

Comparaison code pre-programme 3

Alarme_intrusion_tout.sch Lycee Cournot
2 3



THEME 2010 Validation code FP11

Commande sirene

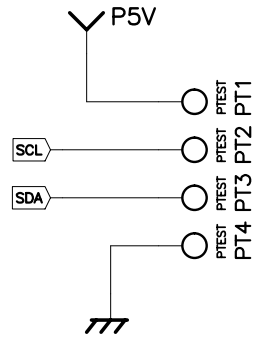
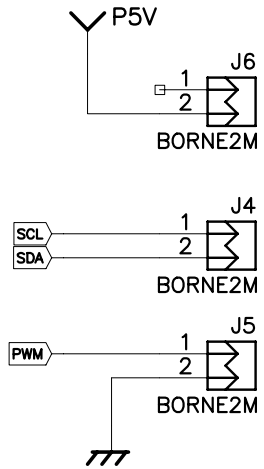
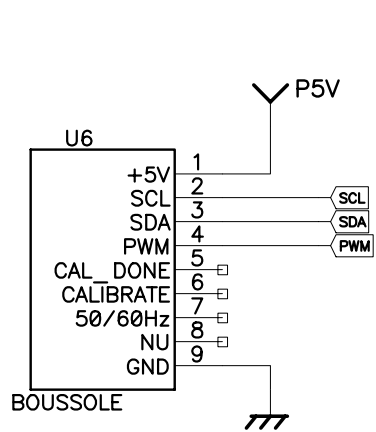
3

Date
Alarme_intrusion_tout.sch

Lycee Cournot
3 3

4.3. FP21

FP21



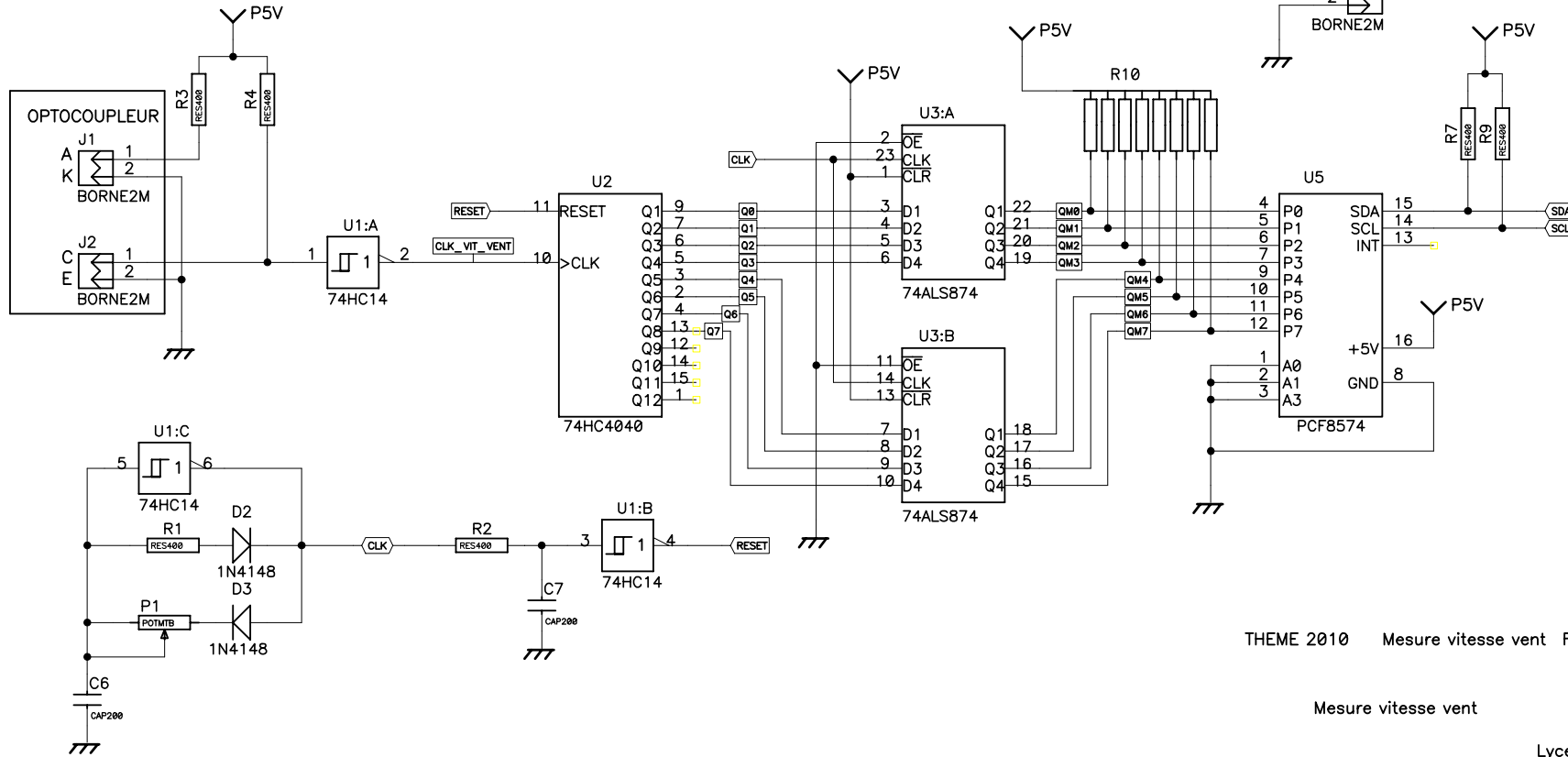
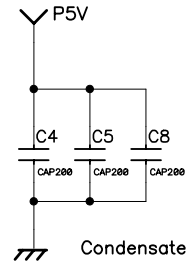
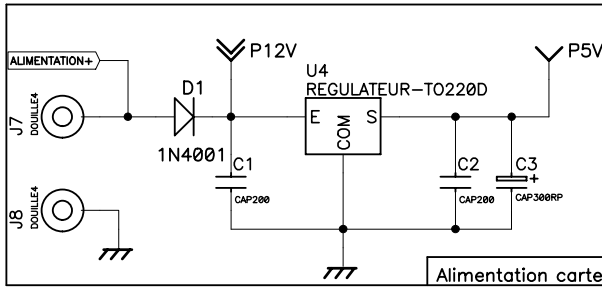
THEME 2010 Mesure orientation vent FP21

Mesure orientation vent 1

Date Lycee Cournot
ORIENTATION_VENT.sch 1 1

4.4. FP22

FP22



THEME 2010 Mesure vitesse vent FP22

Mesure vitesse vent

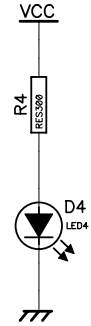
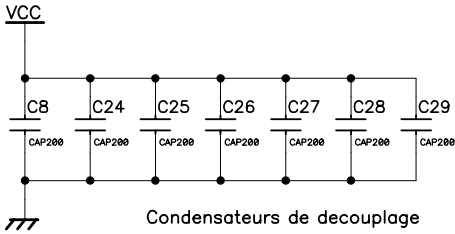
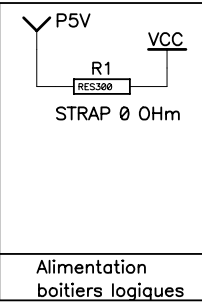
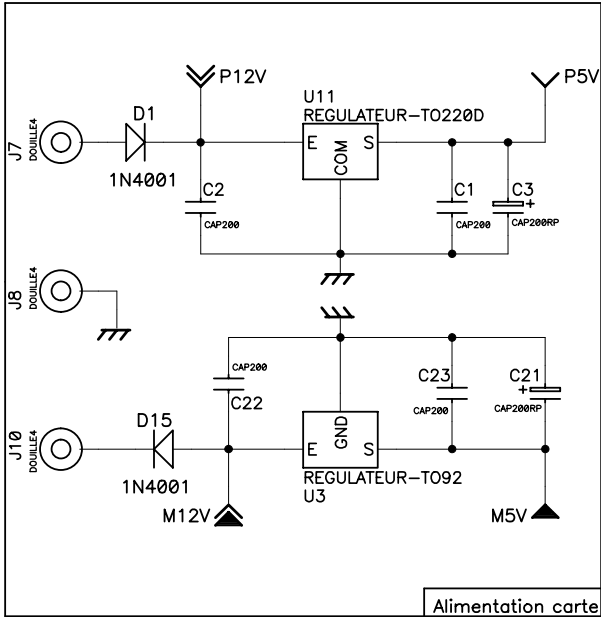
2

Lycee Cournot

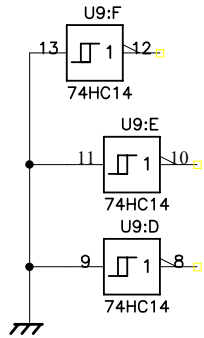
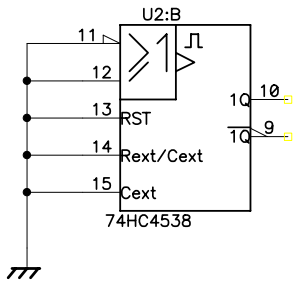
VITESSE_VENT.sch

1 1

4.5. FP3



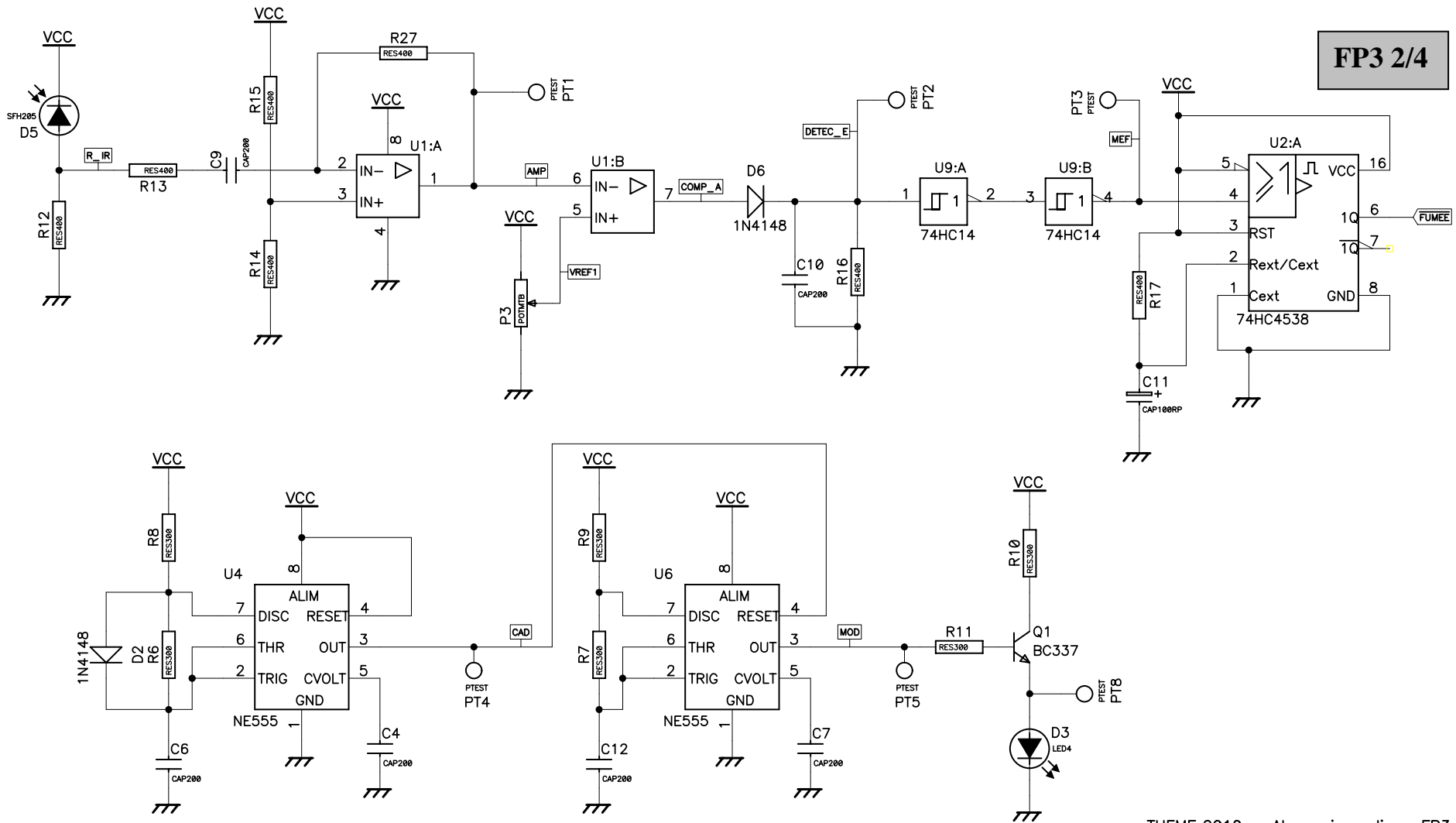
Resistances: prendre RES400 ou exceptionnellement RES300
 Condensateurs: gabarit fonction de la valeur



THEME 2010 Alarme incendie FP3

Alimentation – commande moteur 3
 aspiration – Composants libres

Lycee Cournot
 Alarme_incendie.sch 1 4

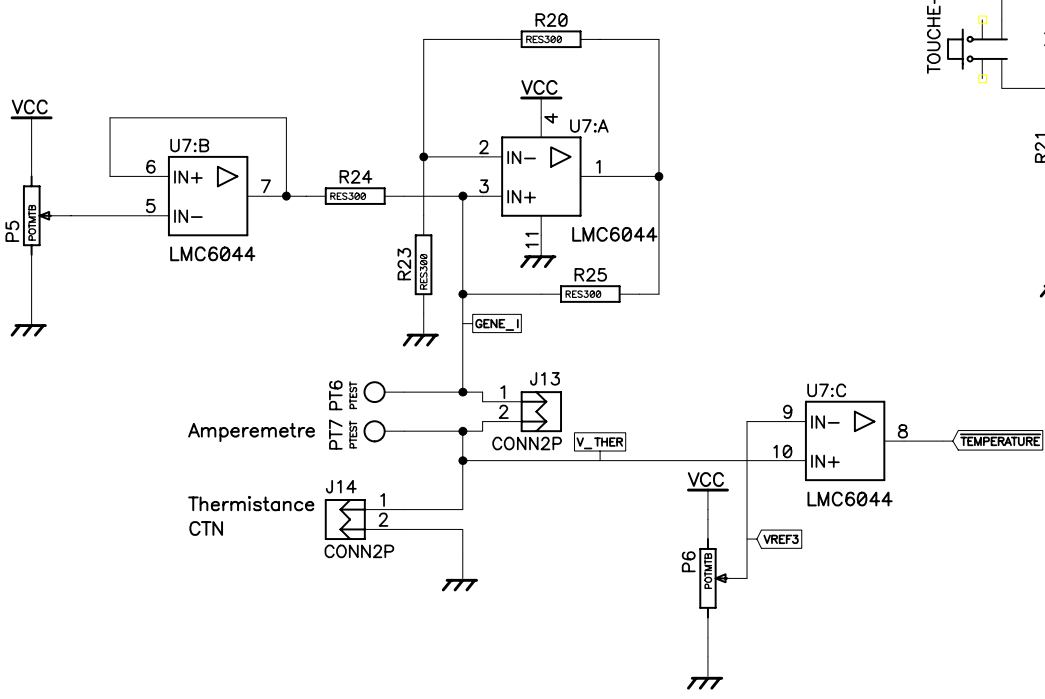
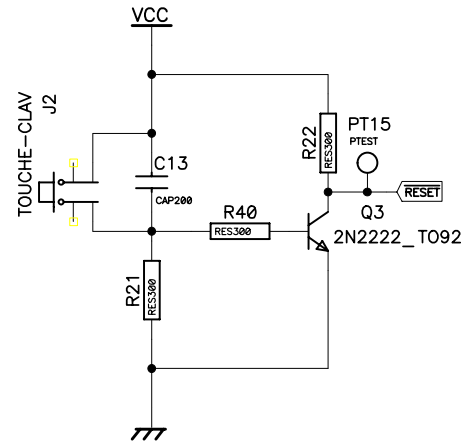
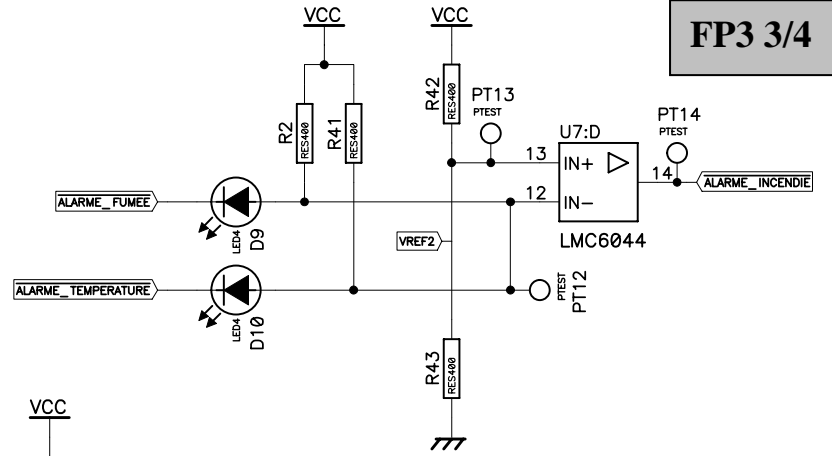
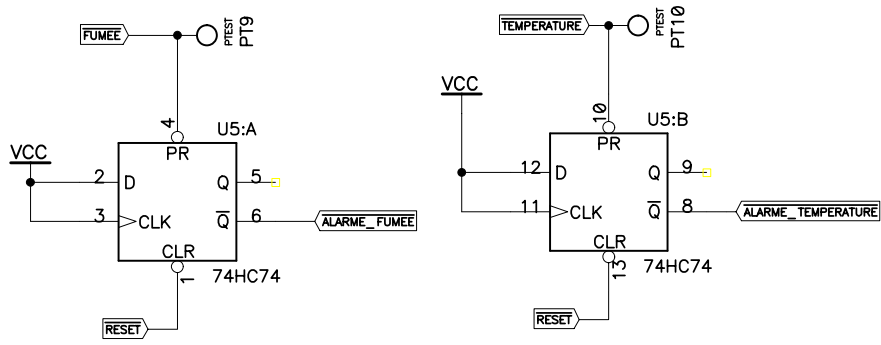


FP3 2/4

THEME 2010 Alarme incendie FP3

Emission Reception infra rouge 3

Alarme_incendie.sch Lycee Cournot
2 4

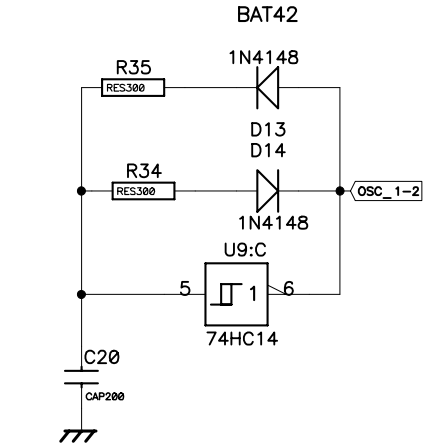
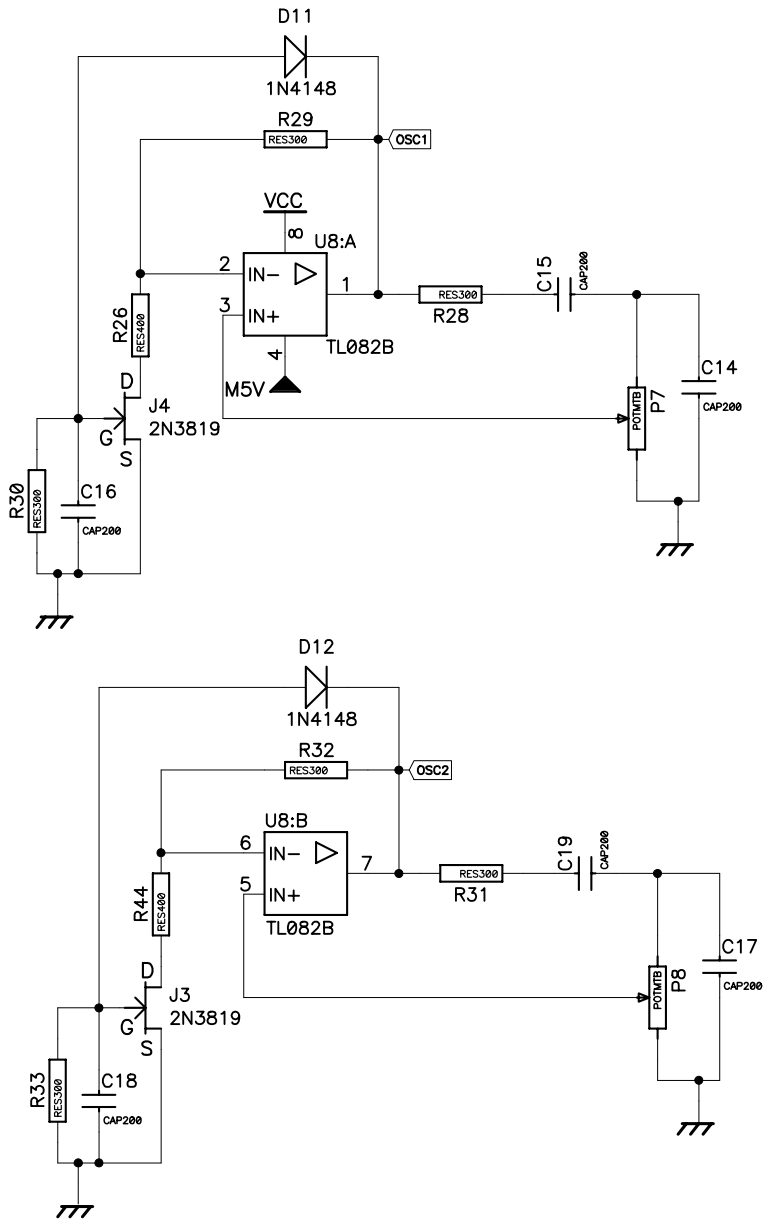


FP3 3/4

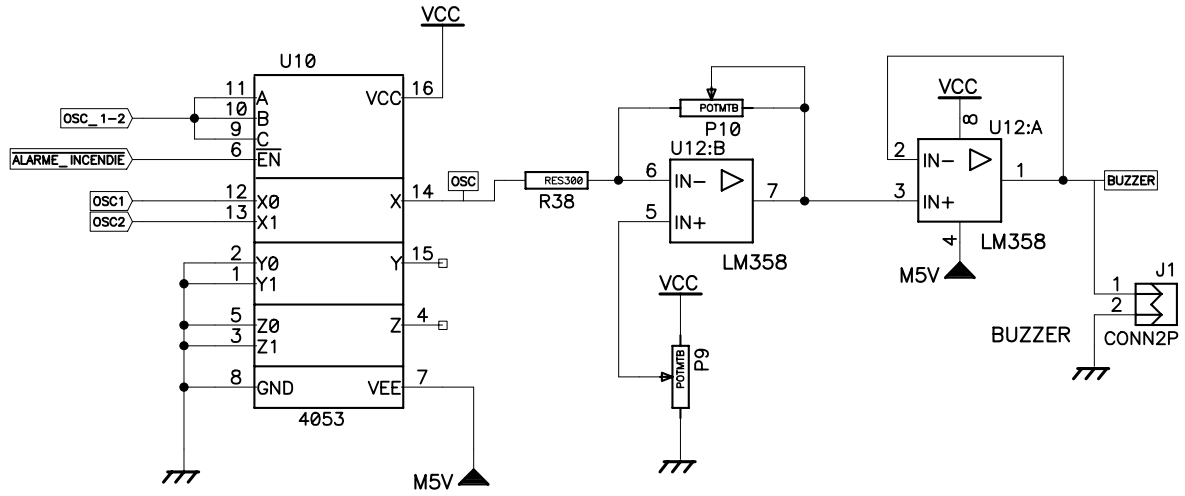
THEME 2010 Alarme incendie FP3

CTN – Alarme 3

Lycee Cournot
Alarme_incendie.sch 3 4



FP3 4/4



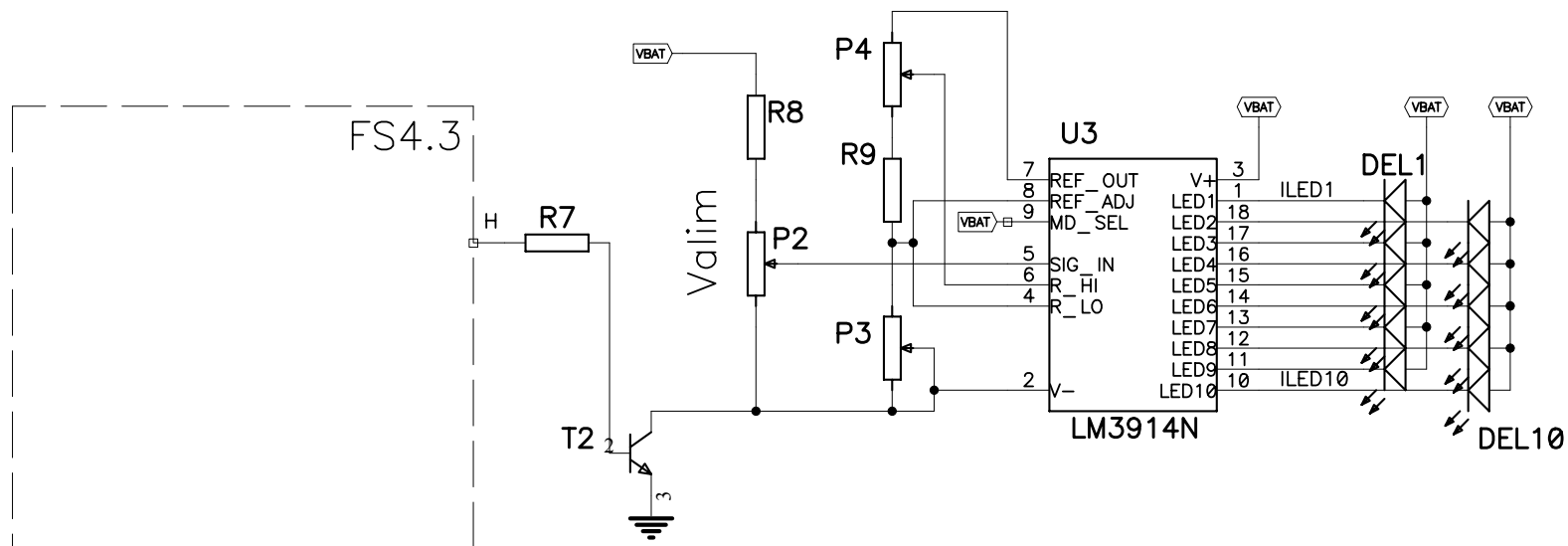
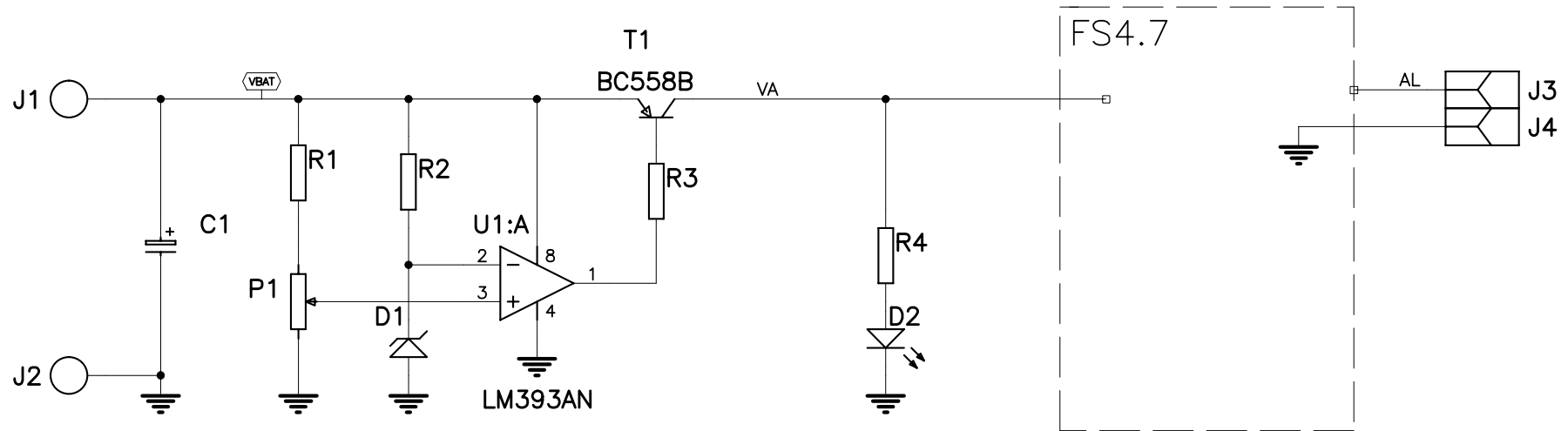
THEME 2010 Alarme incendie FP3

Sirene 2 tons 3

Date Lycee Cournot
 Alarme_incendie.sch 4 4

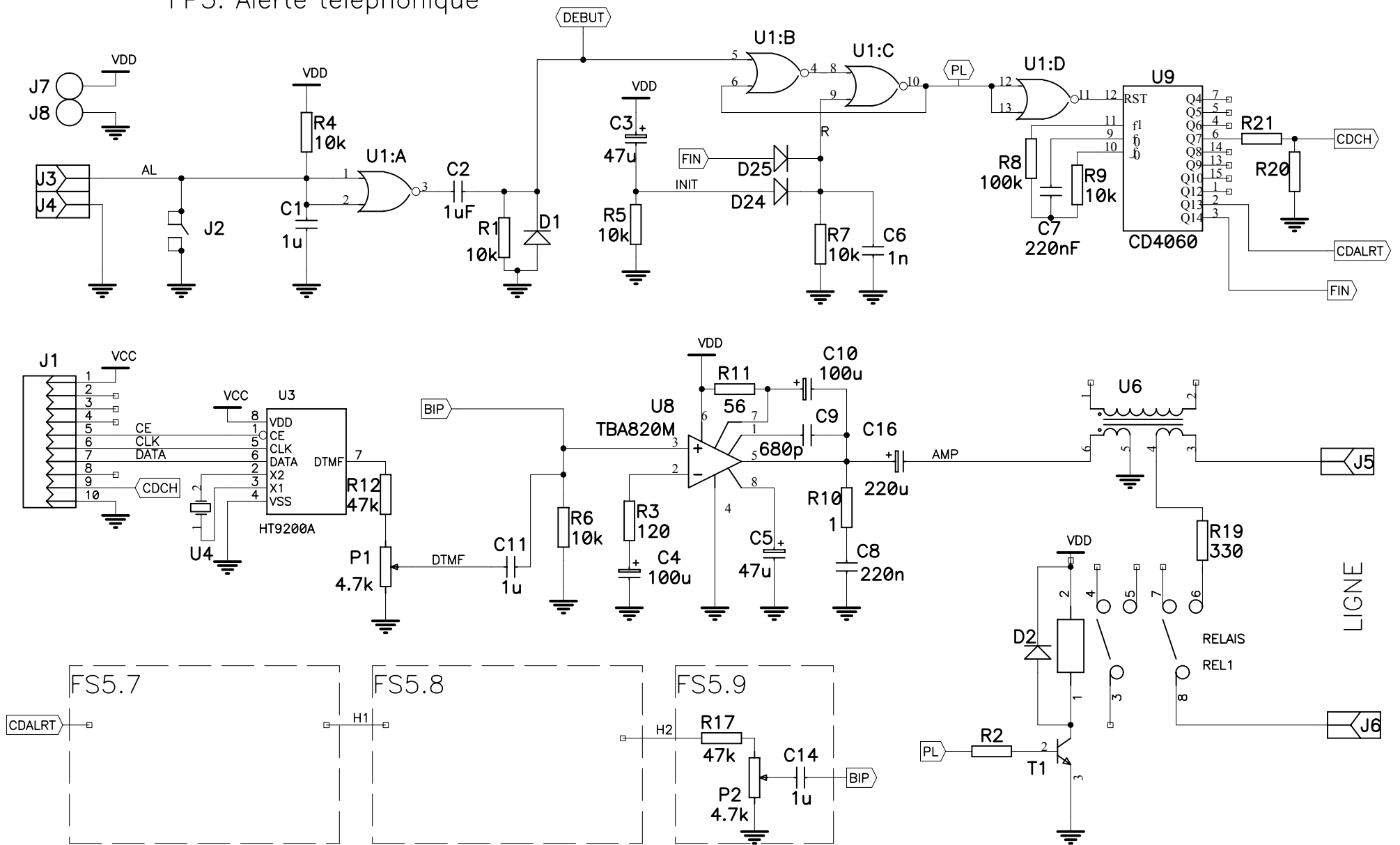
4.6. FP4

FP4:Controle du niveau de la batterie

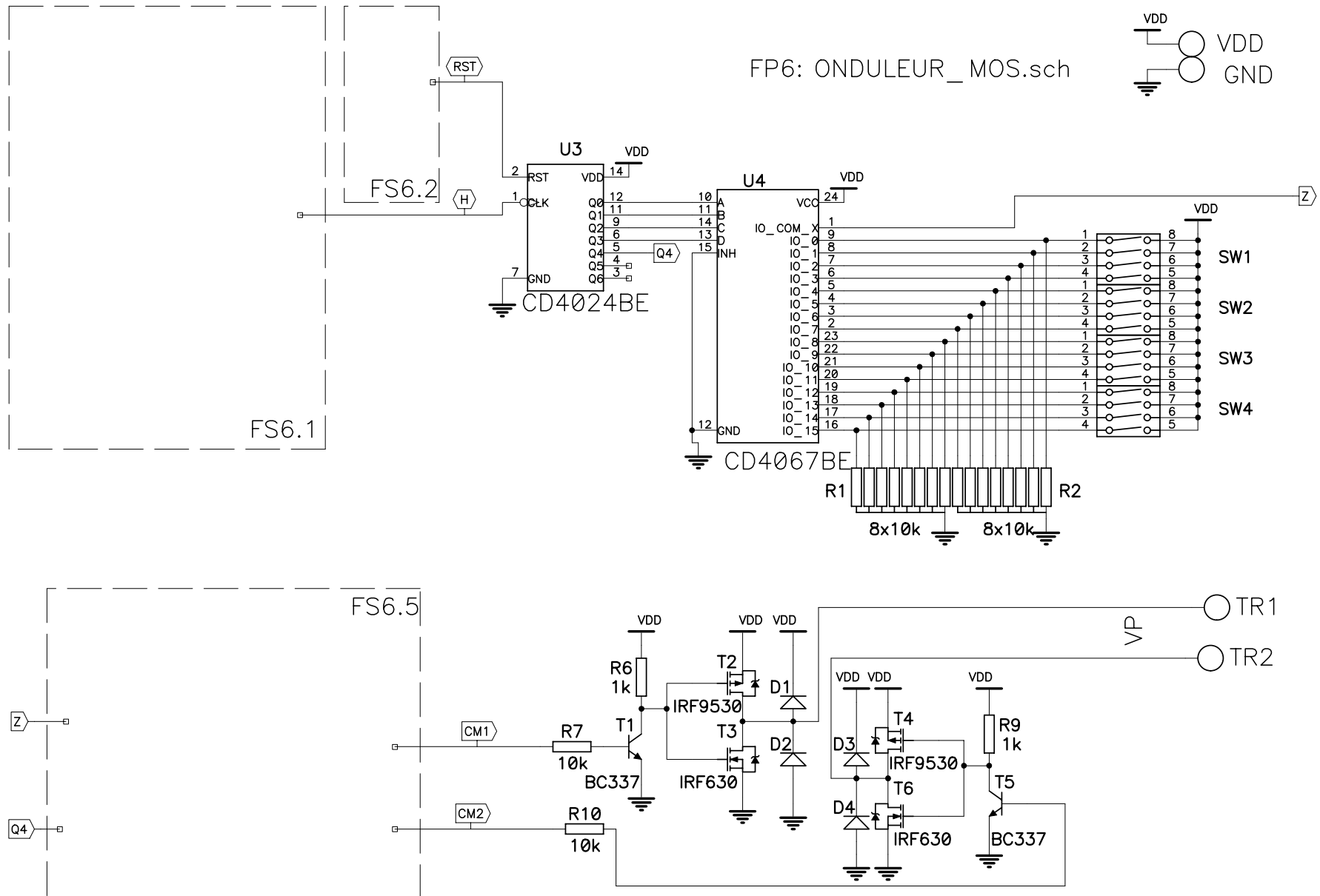


4.7. FP5

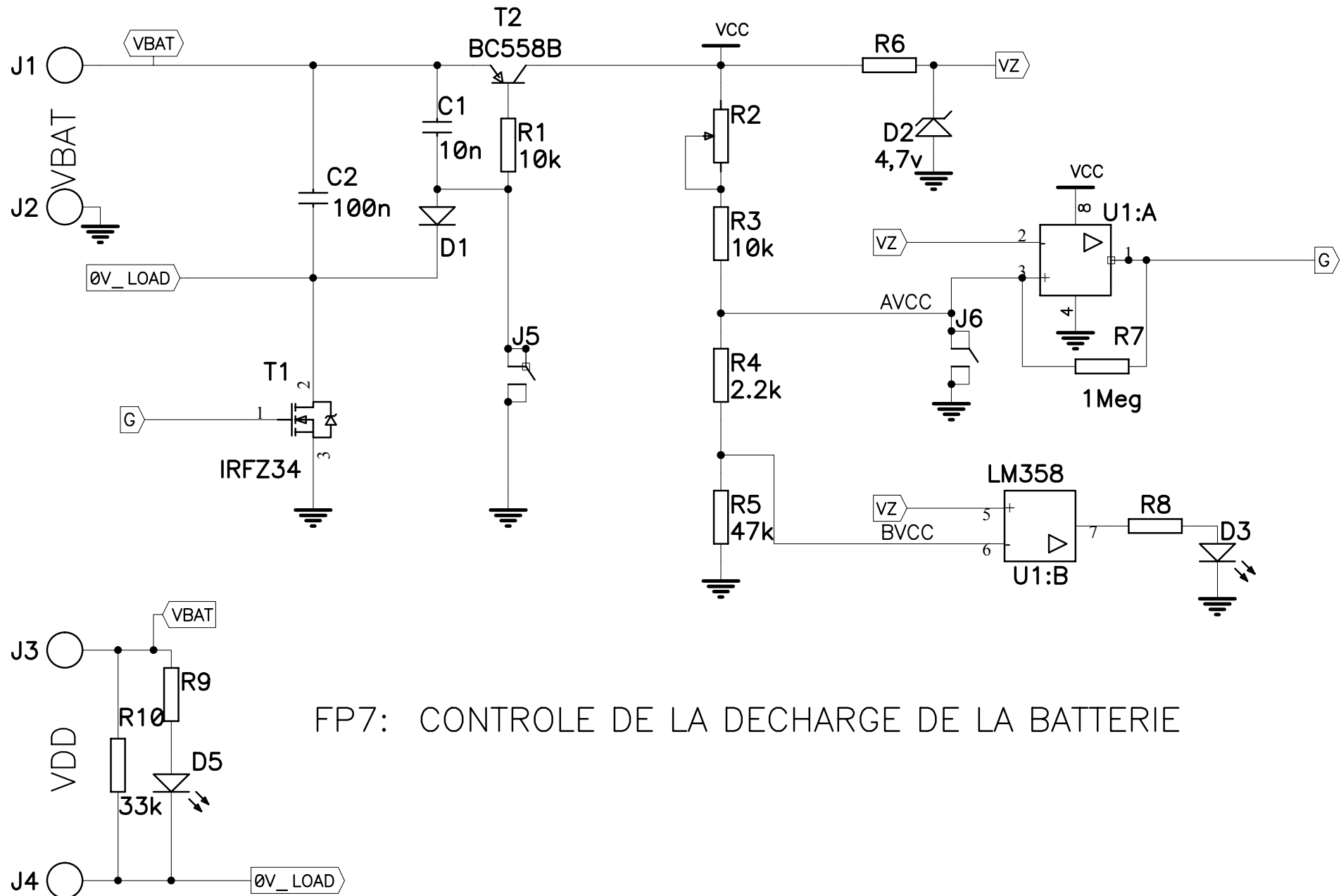
FP5: Alerte telephonique



4.8. FP6



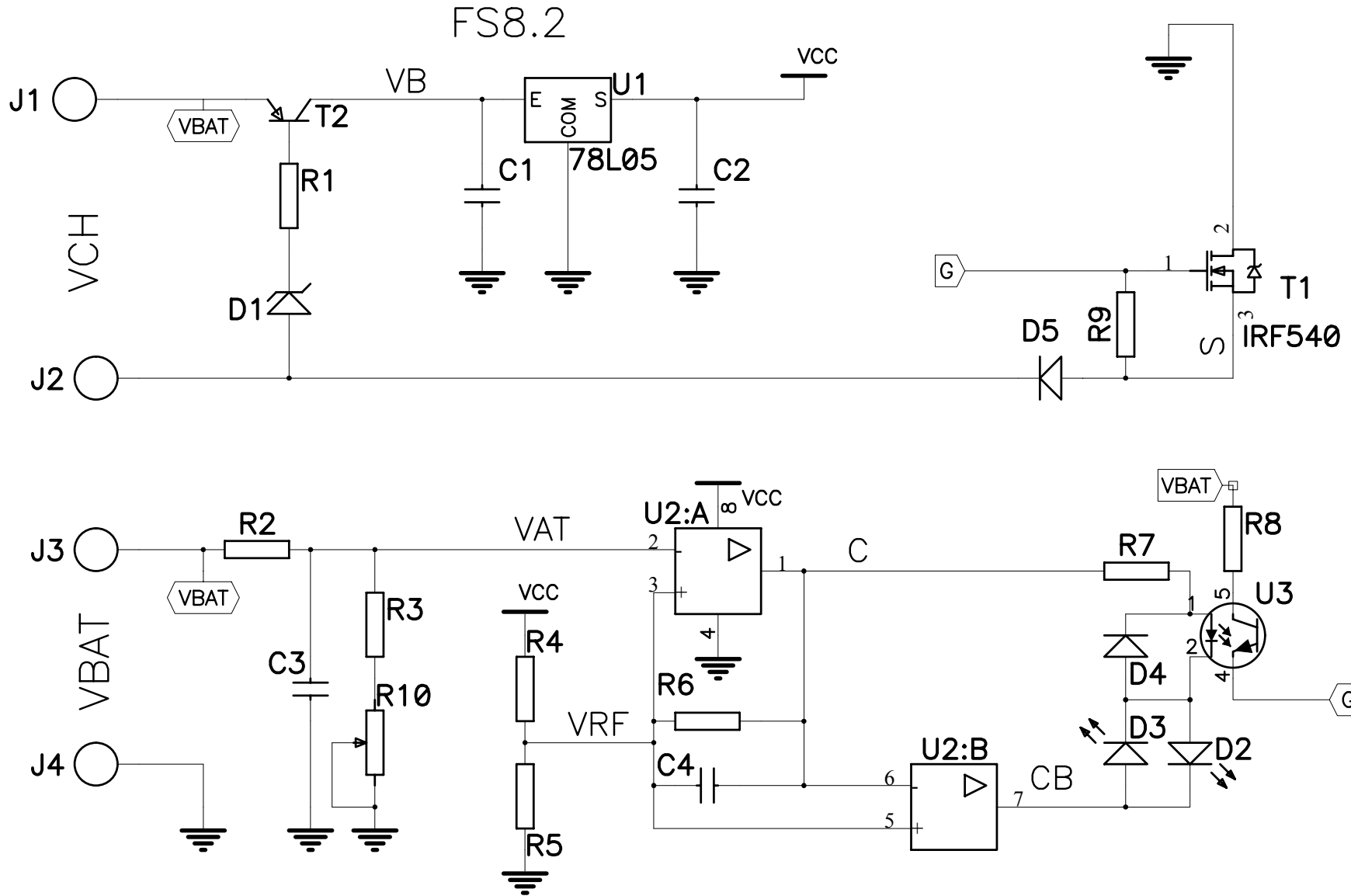
4.9. FP7



FP7: CONTROLE DE LA DECHARGE DE LA BATTERIE

4.10. FP8

FP8: Controle de charge de la batterie



4.11. Nomenclature

FP10

Repère	Valeur	Repère	Valeur	Repère	Valeur
C1	1nF	C9	100nF	R17	NM (2)
C10	470µF	D1	NM	R18	
C13	100nF	D2	1N4001	R19	
C14	100nF	D3	NM	R2	120k
C15		D4	LED 3mA rouge	R3	120k
C16	100nF	J3	IRA E700 (1)	R4	8,2M
C17	100nF	P1		R5	1k
C2	1nF	P2		R8	100k
C3	4,7µF	P3	100k	R9	3,9M
C4	NM	Q1	BC337	U1	LM358
C5	2,2nF	R1	100k	U10	78L05 TO220
C6	NM	R13		U2	LM393
C7	2,2nF	R14		U3	NE555
C8	3,3µF	R16			

- (1) L'IRA E700 ne supportant une température de soudage maximale de 255° pendant 3s, celui-ci sera impérativement monté sur un support type TO39. Composant sensible aux ESD (Electro Static Discharge).
- (2) NM : Non Montée

FP11

Repère	Valeur	Repère	Valeur	Repère	Valeur	Repère	Valeur
C1	100nF	C8		R16		U11	74HC14
C10		C9	100nF	R17		U12	74HC4040
C11	100nF	D1	1n4001	R18	47K	U14	74HC193
C12	100nF	D2	1N4148	R19	470k	U15	LM358
C13	100nF	D3	LED 3mA verte	R2	10k	U2	74HCT688
C14	100nF	J10	DIPSWITCH 8	R20	470k	U20	NE555
C15	100nF	J11	DIPSWITCH 8	R22	10k	U3	74HCT688
C16	100nF	P1		R23	NM	U4	74HC139
C17	1µF	Q2	2N2222	R3	NM	U5	74C922
C18		Q3	2N2907	R4	1k	U6	74ALS874
C19	BUZZER	R1	1k	R5		U7	74HC02
C2	1µF	R10	R-SIL 4,7k	R6		U8	74HC74
C20	220nF	R11	10k	R7		U9	74HC4046
C3	470µF	R12	220k	R8	R-SIL 4,7k		
C4	100nF	R13		R9			
C5	3,3µF	R14	NM	U1	74ALS874		
C6		R15	1k	U10	78L05		

FP22

Repère	Valeur	Repère	Valeur
C1	100nF	R1	
C2	100nF	R10	4,7k
C3	470µF	R2	
C4	100nF	R3	
C5	100nF	R4	
C6		R7	4,7k
C7		R9	4,7k
C8	100nF	U1	74HC14
D1	1N4001	U2	74HC4040
D2	1N4148	U3	74ALS874
D3	1N4148	U4	78L05 TO220
P1		U5	PCF8574

FP3

Repère	Valeur	Repère	Valeur	Repère	Valeur	Repère	Valeur
C1	100nF	C8	100nF	Q1	BC337	R33	39k
C10		D1	1N4001	Q3	2N2222	R34	
C11		D10	LED 3mA	R10		R35	
C12		D11	1N4148	R11		R38	10k
C13	10µF	D12	1N4148	R12	100k	R4	1k
C14	4,7nF	D13	1N4148	R13		R40	10k
C15	4,7nF	D14	1N4148	R14		R41	
C16	680nF	D15	1N4001	R15		R42	22k
C17	10nF	D16	1N4148	R16		R43	33k
C18	680nF	D17	1N4148	R17		R44	820
C19	10nF	D2	1N4148	R20		R5	NM
C2	100nF	D3	LD 274	R21	1k	R6	
C20		D4	LED 3mA rouge	R22	47k	R7	
C21	470µF	D5	SFH205	R23		R8	
C22	100nF	D6	1N4148	R24		R9	
C23	100nF	D9	LED 3mA jaune	R25		U1	LM358
C24	100nF	J14	CTN 2,2k	R26	820	U10	4053
C25	100nF	J3	2N3819	R27		U11	78L05 TO220
C26	100nF	J4	2N3819	R27		U12	TL082
C27	100nF	P10	47k	R28	10k	U2	74HC4538
C28	100nF	P3	10k	R29	2,2k	U3	79L05 TO92
C29	100nF	P5	10k	R29	2,2k	U5	74HC74
C3	470µF	P6	10k	R3	NM	U7	LMC6044
C4	47nF	P7	47k	R30	27k	U8	TL082
C5	BUZZER	P8	47k	R31	10k	U9	74HC14
C6		P9	10k	R32	2,2k		

FP4 Contrôle niveau batterie

Repère	Valeur	Repère	Valeur	Repère	Valeur	Repère	Valeur
C1	100u	DEL5		P2	4.7k	R7	
D1	4.7v	DEL6		P3	4.7k	R8	4.7k
D2	...	DEL7		P4	1k	R9	4.7k
DEL1		DEL8		R1	...	T1	BC558B
DEL2		DEL9		R2		T2	...
DEL3		DEL10		R3		U1	LM393AN
DEL4		P1	...	R4		U3	LM3914N

FP5 Alerte téléphonique

Repère	Valeur	Repère	Valeur	Repère	Valeur	Repère	Valeur
C1	1u	D1	1N4148	P2	4.7k	R17	47k
C2	1uF	D2	...	R1	10k	R19	330
C3	47u	D24	1N4148	R2	...	R20
C4	100u	D25	1N4148	R3	120	R21	...
C5	47u	J1	10HE10	R4	10k	REL1	
C6	1n	J2	B.P. NO	R5	10k	T1	...
C7	220nF	J3	BORNE	R6	10k	U1	CD4001
C8	220n	J4	BORNE	R7	10k	U3	HT9200A
C9	680p	J5	BORNE	R8	100k	U4	3.579545M
C10	100u	J6	BORNE	R9	10k	U6	1,5VA 2x6
C11	1u	J7	DOUILLE	R10	1	U8	TBA820M
C14	1u	J8	DOUILLE	R11	56	U9	CD4060
C16	220u	P1	4.7k	R12	47k		

FP6 Onduleur

Repère	Valeur	Repère	Valeur	Repère	Valeur	Repère	Valeur
D1	BYV27	R6	1k	SW3	SW_DIP4	T5	BC337
D2	BYV27	R7	10k	SW4	SW_DIP4	T6	IRF630
D3	BYV27	R9	1k	T1	BC337	U3	CD4024BE
D4	BYV27	R10	10k	T2	IRF9530	U4	CD4067BE
R1	8x10k	SW1	SW_DIP4	T3	IRF630		
R2	8x10k	SW2	SW_DIP4	T4	IRF9530		

FP7 Contrôle décharge batterie

Repère	Valeur	Repère	Valeur	Repère	Valeur	Repère	Valeur
C1	10n	J2	DOUILLE	R3	10k	R10	33k
C2	100n	J3	DOUILLE	R4	2.2k	T1	IRFZ34
D1	1N4001	J4	DOUILLE	R5	47k	T2	BC558B
D2	4,7v	J5	B.P. no	R6	U1	LM358
D3	DEL rouge	J6	B.P. no	R7	1Meg		
D5	DEL vert	R1	10k	R8		
J1	DOUILLE	R2	R9		

FP8 Contrôle charge batterie

Repère	Valeur	Repère	Valeur	Repère	Valeur	Repère	Valeur
C1	100n	D3	DEL rouge	R4	10k	R10
C2	100n	D4	1N4148	R5	10k	T1	IRF540
C3	100n	D5	SB530	R6	47k	T2	BC558B
C4	47n	R1	R7	...	U1	78L05
D1	12V	R2	33k	R8	10k	U2	LM358
D2	DEL verte	R3	R9	47k	U3	4N25

Pour chaque fonction (sauf mention spéciale) il est demandé une étude qualitative, quantitative et un relevé expérimental

5. TRAVAUX ELEVES

5.1. Travail groupe 1

On ne demande pas d'étudier FS10.1 et FS10.2 car les niveaux de tensions sont trop faibles. Il est possible de relever le signal de sortie de FS10.2

1. Fabriquer la carte FP11 (le typon est fourni)
2. Dessiner, router et fabriquer FP10
3. FS10.2 réglage P3
4. FS10.3 Déterminer la valeur des composants et régler les potentiomètres
5. FS10.4 Déterminer la valeur des composants
6. FS10.5 Déterminer la valeur des composants
7. FS11.1 Relever un tableau des codes des touches (sortie connecteur J5)
8. FS11.2, FS11.3, FS11.4, FS11.5, FS11.7 vérifier le bon fonctionnement
9. FS11.6 relever le 5V et INIT à la mise sous tension et expliquer le rôle
10. FS11.8 Déterminer R5, P1, C6
11. FS11.9 Déterminer R9, R7, C8
12. FS11.10 Déterminer R13, C10 (R14 non montée)
13. FS11.11 Déterminer R6, C18 de façon à garder la fréquence fondamentale et filtrer les fréquences harmoniques
14. FS11.11 Déterminer R16, R17
15. FP20 (logiciel) : compléter logiciel de programmation du code et d'acquisition du code (utilisation clavier externe et afficheur LCD) suivant l'algorithme suivant. Les fonctions de gestion du clavier et de l'afficheur LCD sont fournies
Fichier programme à compléter : th10_G1eleve.C

```
// Programmation du code d'entrée
//-----
void ProgrammationCodeEntree(void)
{
    char i;

    WriteCmdXLCDb(LIGNE1);
    AffChaine("Prog. code? ");
    WriteCmdXLCDb(LIGNE2);
    AffChaine("OUI->1 NON->2 ");

    while (ToucheAppuyee != OK); //Attendre tant qu'une touche n'est pas appuyée
    ToucheAppuyee = NOK; //Reinitialiser la variable ToucheAppuyee
    if (Clavier[0] == '1') // la touche appuyee est '1'
    {
        WriteCmdXLCDb(LIGNE1);
        AffChaine("Code 4 touches: ");
        WriteCmdXLCDb(LIGNE2);
        AffChaine("____ ");
        MettreBlanc();
        WriteCmdXLCDb(LIGNE2); // reinitialiser curseur en debut de ligne

//A COMPLETER DEBUT
//-----
// boucle de programmation du code (4 touches)
        for
        {
//Attendre tant qu'une touche n'est pas appuyée

//Reinitialiser la variable ToucheAppuyee

//Mettre dans le tableau ProgrammationCode le code de la touche appuyee

//remplacer les '_' par des '*' au fur et à mesure de la saisie du code

//A COMPLETER FIN
//-----
        }
        }
    _delay_ms(500);
    WriteCmdXLCDb(LIGNE1);
    AffChaine("...");
    MettreBlanc();
    ClearLine(LIGNE2);
}
```

5.2. Travail groupe 2

1. La carte FP20 est fournie complète
2. Fabriquer la carte FP21 (le typon est fourni)
3. Dessiner, router et fabriquer FP22
4. Configuration du module EZL (vitesse RS232 et adresse IP). L'adresse IP fixe est à demander à l'administrateur réseau
5. FP20 logiciel :
 - a. A l'aide du cours I2C, des fonctions I2C fournies et de la documentation du PCF8574, écrire la fonction `char VitesseVent(void)` permettant l'acquisition d'une valeur hexadécimale proportionnelle à la vitesse du vent suivant l'algorithme suivant :

```
// Acquisition vitesse vent
//-----
char VitesseVent(void)
{
    char resultat;

    // Start I2C

    // Écrire l'adresse I2C du PCF8724, mode lecture

    // Lire le contenu du PCF 8724 (stockage dans la variable resultat)

    // Stop I2C

    return resultat;
}
```

- b. Travail préliminaire à l'écriture de la fonction `char AcquisitionTensionBatterie(void)`
La batterie délivre une tension maximale de 18V en pleine charge. On définit 3 états de fonctionnement en fonction de sa tension:
 - 18 -13 V : Pleine charge
 - 11 – 13 V : charge faible
 - 10 – 11 V : Décharge
 - Inférieure à 10V : batterie hors service

Cette tension ne peut pas entrer directement sur l'ADC de l'ATMEGA32 qui est limité à 0 – 5V.

- i. Déterminer les valeurs d'un pont résistif permettant de passer de 14V à 5V. La sortie de ce pont sera reliée à l'entrée de l'ADC canal 3 de l'ATMEGA32 sur laquelle on placera 2 diodes de protection à 5V et 0V (diodes de clamping).
 - ii. Déterminer alors les valeurs caractéristiques en Volts correspondant aux seuils des 4 états définis ci-dessus
 - iii. Déterminer les valeurs numériques correspondantes en sortie de l'ADC (ADC 8bits, tension maximale 5V, tension minimale 0V)
- c. Ecrire la fonction `char AcquisitionTensionBatterie(void)` suivant l'algorithme suivant :

```
// Acquisition tension batterie
//-----
char AcquisitionTensionBatterie(void)
{
    char TCB;

    // Selectionner le canal 3 de l'ADC

    // Lancer la conversion

    // Attendre que la conversion soit finie

    // Si tension batterie inférieure à 14V TCB = 3
    // Si tension batterie inférieure à 12.5 V TCB = 2
    // Si tension batterie inférieure à 11 V TCB = 1
    // Si tension batterie inférieure à 10 V TCB = 0

    // Afficher le résultat de la conversion sur 8 LEDs (ou l'afficheur LCD)

    return TCB;
}
```

d. Ecrire la fonction `int OrientationVent(void)` suivant l'algorithme suivant :

```
// Acquisition orientation eolienne
//-----
int OrientationVent(void)
{
    char reg;
    int orientation;
    float calcul ;

// Orientation sur 1 octet
//=====

// Ecriture du numéro du registre contenant la valeur de l'orientation
//-----
// Start I2C

// Écrire l'adresse I2C du CMP03, mode ecriture

// Ecrire (data) le numéro du registre contenant la valeur de l'orientation sur 1 octet

// Lecture du registre contenant la valeur de l'orientation
//-----
// Start I2C

// Écrire l'adresse I2C du CMP03, mode lecture

// Lire le contenu du CMP03 (stockage dans variable reg)

// Stop I2C

// Fin de la transmission I2C
//-----
// Mise à l'échelle 0-360°:
// * variable reg varie de 0 à 255
// * on veut stocker dans la variable calcul une variation de 0 à 360 suivant la
// variation de reg

// Prendre la partie entière de calcul

// Renvoyer orientation;
}
```

Fichier programme à compléter : th10_G2eleve.C

6. FS22.1 Déterminer R3 et R4
7. FS22.5 Déterminer la valeur des composants R1, P1, C6
8. FS22.6 Déterminer la valeur des composants R2, C7
9. FP22 Relever une trame I2C donnant le nombre d'impulsions mesurée

5.3. Travail groupe 3

On ne demande pas d'étudier FS3.4 car les niveaux de tensions sont trop faibles.

1. Dessiner, router et fabriquer FP3
2. FS3.1 Déterminer R6, R8, C6
3. FS3.2 Déterminer R7, R9, C12
4. FS3.3 Déterminer R10 pour avoir un courant de LED de 25mA Déterminer R11
5. FS3.5 Déterminer R27 et R13 (en petits signaux C9 est assimilable à un court circuit). Déterminer R14 R15.
6. FS3.6 Pour étudier cette fonction il faut simuler la présence de fumée entre la LED émettrice infra rouge et la photodiode réceptrice. Ceci sera réalisé en pliant une feuille de calque en 2 (représentant ainsi l'opacité de la fumée) et placée entre émetteur et récepteur. Régler V_{REF1} pour que COMP_A soit à 0 lors de la présence de fumée.
7. FS3.7 Déterminer R16 C10
8. FS3.9 Déterminer R17 C11
9. FS3.10, FS3.12, FS3.14 Vérifier le bon fonctionnement
10. FS3.13 En considérant $R24=R23=R20=R25$, déterminer ces résistances pour avoir un générateur de courant de 1mA
11. FS3.15 Déterminer la valeur de V_{THER} pour une température de 33°C. Régler VREF3 pour satisfaire ce seuil
12. FS3.16, FS3.18 Vérifier le bon fonctionnement, tracé de bode pour le filtre seul (fréquence centrale, atténuation), calcul de l'amplification pour satisfaire la condition d'oscillation et réglage.
13. FS3.17 Déterminer R34, R35, C20
14. FS3.19 Vérifier le bon fonctionnement

15. FS3.21 Relever le 5V et INIT à la mise sous tension et expliquer le rôle

16. FS3.20 Logiciel

- a. Calculer la valeur numérique N_{25} (CAN) d'une température de 25°. De même pour N_{33} , une température de 33°. Cette dernière valeur sera le seuil de détection au dessus duquel on considèrera que la température est anormalement élevée (ce n'est pas réaliste, mais cette température s'obtient facilement en chauffant peu la thermistance)
- b. Ecrire la fonction `char AcquisitionTemperatureAmbiante(void)` permettant d'acquérir la valeur numérique de la température mesurée par la CTN et de conclure sur l'état (température excessive ou non)

```
// Acquisition température ambiante
//-----
char AcquisitionTemperatureAmbiante(void)
{
    char Temp = NOK;

    // Selectionner le canal 0 de l'ADC
    // Lancer la conversion
    // Attendre que la conversion soit finie (resultat N)
    // Si N est supérieur à N33 Temp = OK
    // Sinon Temp = NOK
    // Afficher la valeur de Temp sur LED
    // Renvoyer N
    // Afficher le résultat de la conversion N sur l'afficheur LCD
    // (Peut être mis dans le programme principal)
```

Fichier programme à compléter : th10_G3eleve.C

5.4. Travail groupe 4

FP4

1. Choix technologique de R2, R1 et P1.
2. Choix technologique de R3 et R4
3. Synthèse de FS4.7
4. Synthèse de FS4.3
5. Calcul de Icc
6. Procédure de réglage de FS4.5
7. Chronogrammes des signaux de FP4 lorsque VBAT passe de 13v à 8v

FP8

8. Choix technologique de R1 (T2 et D1 donnés)
9. Choix technologique de R3 et R10 de façon à pouvoir régler le seuil de déconnection de la batterie de 13v à 15v
10. Calcul des seuils pour FS8.5
11. Choix de R7
12. Chronogrammes des signaux de FP8 lorsque VBAT passe de 13v à 18v

5.5. Travail groupe 5

FP5

1. Calcul de la durée de l'impulsion en sortie de FS5.1
2. Calcul de la durée de l'impulsion sur INIT
3. Calcul de durées TCDCH, TCDALRT et TFIN
4. Justification du choix de R3 et C9
5. Synthèse de FS5.7 et FS5.8
6. Choix de R2

7. Chronogrammes des signaux de FP5
8. Adapter le programme fourni afin de générer sur DTMF le chiffage du numéro "0384650701" (le chiffage doit durer environ trois secondes et être conforme à la documentation technique du HT9200A)

```

/*-----FP5.c-----
    PA3: CE          PA2:CDCH
    PA4: CLK
    PA5: DATA
-----*/
#define F_CPU 8000000UL      /* freq horl: pour les fonctions _delay */
#include <avr/io.h>          /* etiquettes PORTA, DDRA etc..*/
#include <util/delay.h>      /* fonctions _delay_ms et _delay_us */
//-----envoi d'un chiffre-----
void digit (unsigned char n)
{
    unsigned char i;
    for (i=0;i<5;i++)
        {
            if ((n & 0b00000001)==1)
                {
                    PORTA |= 0b00100000;
                }
            else
                {
                    PORTA &= 0b11011111;
                }
            _delay_us (...);          // -----à compléter-----
            PORTA &= 0b11101111;
            _delay_us(...);          // -----à compléter-----
            PORTA |= 0b00010000;
            n=n>>1;
        }
    _delay_ms (...);                // -----à compléter-----
}

// -----chiffage d'un numero-----
void Chiffrage(char *numero)
{
    PORTA &= 0b11110111;
    _delay_ms(...);                // -----à compléter-----
    while (*numero != '\0')
        {
            if (*numero==48) digit (10); /* pour "0": envoyer le code 10
                                           (doc.HT9200A page 5 Tble 1)*/
            else digit(*numero-48);     /* sinon: transcoder
                                           code ASCII --> valeur numerique*/
            numero++;
        }
    digit (0b00011111);              //DTMF off
    PORTA |= 0b00001000;
}

//=====fonction principale=====
int main(void)

```



```

{
  unsigned char t;
  PORTA=0b11111111;
  DDRA=...; // -----à compléter-----

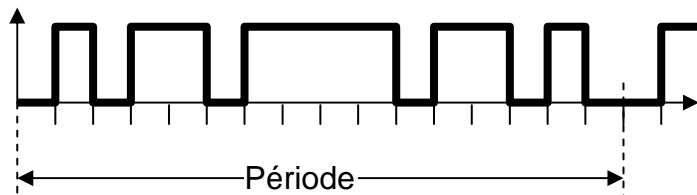
  while (1)
  {
    if ((PINA & 0b00000100) !=0)
    {
      Chiffre ("0123456789");
      for (t=0;t<...;t++) // -----à compléter-----
      {
        _delay_ms(1000);/* ne pas rechiffrer
                          pendant l'appel telephonique*/
      }
    }
  }
}

```

5.6. Travail groupe 6

FP6

1. Synthèse de FS6.1 (prévoir un réglage précis de la fréquence de H)
2. Synthèse de FS6.2
3. Déterminer l'état des switches I0 à I15 permettant d'obtenir le signal suivant sur Z:



4. Synthèse de FS6.5
5. Chronogrammes des différents signaux sur une période de VP

FP7

6. Choix de R6, R8 et R9
7. Calcul et choix de R2 pour pouvoir régler le seuil de déconnection de la charge entre VBAT= 9v et VBAT=11v
8. Calcul de la valeur de VBAT entraînant l'allumage de D3 dans le cas où la déconnection de la charge intervient à VBAT=10v.
9. Chronogrammes des différents signaux de FP7

5.7. Plan du rapport :

Le rapport devra comporter environ 25 pages hors annexe.

Au début, Il devra contenir obligatoirement le cahier des charges du groupe (le travail demandé).

En annexe, ne pourront figurer que les extraits des documents constructeurs nécessaires à la compréhension du rapport (12pages maximum).

Il devra comporter un sommaire et les pages devront être numérotées.

Le rapport pourra suivre le plan suivant:

La partie présentation n'apparaît pas dans le dossier mais doit être parfaitement connue pour l'épreuve orale.

1. Étude fonctionnelle de 1^{er} degré des objets techniques.

- Schémas fonctionnels de 1^{er} degré.
- Explications des fonctions principales.
- Définitions des liaisons.

2. Explications à propos des fonctions étudiées.

- Position et justification de la présence des fonctions au sein du système ;
- Schéma fonctionnel de 2nd degré des fonctions principales ;
- Schémas structurels et nomenclatures ;
- Définitions des liaisons ;
- Étude détaillée de chaque fonction secondaire qui peut comporter par exemple :
 - Schéma structurel de la fonction secondaire ;
 - Explications du fonctionnement de la fonction secondaire ;
 - Calcul ou justification des composants ;
 - Définitions des points tests ;
 - Chronogrammes théoriques et/ou oscillogrammes ;
 - Algorithme de fonctionnement ;
 - Programme de test ;
 - Etc...
 - Méthode de mise en œuvre des cartes ;
 - Relevés des mesures.

3. Algorithme et programmation des cartes étudiées.

4. Documents de fabrication.

- Schémas structurels (si réalisés par le binôme) et nomenclatures chiffrées.
- Typons avec identification des faces (si réalisés par le binôme) et schémas d'implantation.
- Plan de câblage (définition de la connectique).

6. DEFINITION DES FONCTIONS LOGICIELLES FOURNIES

6.1. Fichiers à inclure

`#include <iom32phv.h>`

Description des registres ATMEGA32 (modifié)

`#include <avr/pgmspace.h>`

Permet de pouvoir déclarer des tableaux en FLASH

`#include <avr/interrupt.h>`

Permet d'utiliser les vecteurs d'interruptions

`#include <avr/delay.h>`

Permet d'utiliser les fonctions d'attente

`#include <def_ports.h>`

Permet la commande et le test individuels des bits d'un registre

`#include <AfficheurLCD.h>`

Permet d'utiliser toutes les fonctions de gestion de l'afficheur LCD (voir le paragraphe dédié « Afficheur LCD »)

Avant cet « include », il faut indiquer le port auquel est relié l'afficheur par une instruction de type :

`#define LCD_PORTB` (ce qui signifie ici que l'afficheur est relié au PORTB)

`#include <i2c.h>`

Permet d'utiliser toutes les fonctions de gestion du bus I2C

`#include <pageHTMLh>`

Contenu (codes ASCII) de la page HTML

`#include <TH10_fonctions_G1.h>`

Contient l'ensemble des fonctions nécessaires pour le groupe 1 (faire de même pour groupe 2 et 3)

`#include <TH10_interruptions.h>`

Contient l'ensemble des fonctions lors de l'exécution d'une interruption

6.2. Afficheur LCD

`void WriteCmdXLCDb(unsigned char cmd);`

Sélectionne la ligne de l'afficheur sur laquelle on désire écrire

Exemple : `WriteCmdXLCDb(LIGNE1);`

Permet de sélectionner la première ligne (rien n'est encore écrit) et de positionner le curseur (visible ou non en début de ligne).

`void AffChaine(prog_char *TabChaine);`

Permet l'affichage d'une chaîne constante

Exemple :

`WriteCmdXLCDb(LIGNE2);`

`AffChaine("Bonjour");`

Affichage de **Bonjour** sur la ligne 2

`void AffVarChaine(char *TabChaine);`

Permet l'affichage d'une chaîne contenue dans une variable

Exemple :

`char *tab[2] = {"Bonjour", "Au revoir"};`

`WriteCmdXLCDb(LIGNE1);`

`AffVarChaine(tab[0]);`

```
WriteCmdXLCdb(LIGNE2);
AffVarChaine(tab[1]);
```

Affichage de **Bonjour** sur la ligne 1
Affichage de **Au revoir** sur la ligne 2

void AffEntier(long N);

Permet d'afficher la valeur entière de N

Exemple :

```
N=32 ;
WriteCmdXLCdb(LIGNE1);
AffChaine("N=");
AffEntier(N);
```

Affichage de **N=32** sur la ligne 1

void AffHexa(long N);

Permet d'afficher la valeur hexadécimale de N

Exemple :

```
N=31 ;
WriteCmdXLCdb(LIGNE1);
AffChaine("N=");
AffHexa(N);
```

Affichage de **N=h1F** sur la ligne 1

void AffReel(float N, char ChiffreApresVirgule);

Permet d'afficher la valeur réelle de N

Exemple :

```
U=9.85 ;
WriteCmdXLCdb(LIGNE1);
AffChaine("U=");
AffReel(U,2); // 2 -> nombre de chiffres après la virgule
AffChaine(" V");
```

Affichage de **U=9.85 V** sur la ligne 1

void ClearLine (char ligne);

Efface la ligne sélectionnée

Exemple : ClearLine(LIGNE1);

Efface la ligne 1

void MettreBlanc(void);

Efface la fin de ligne en cas de réécriture sur la même ligne sans effacement

Exemple :

```
WriteCmdXLCdb(LIGNE1);
AffChaine("*** Bonjour ***");
...
WriteCmdXLCdb(LIGNE1);
AffChaine("Choix?");
```

Donnera l'affichage **CHOIX?njour *****

Alors que

```
WriteCmdXLCdb(LIGNE1);
AffChaine("*** Bonjour ***");
...
WriteCmdXLCdb(LIGNE1);
AffChaine("Choix?");
MettreBlanc(void);
```

Donnera l'affichage **CHOIX?**

6.3. Bus I2C

void I2Cstart(void);

Permet d'envoyer la séquence de start sur le bus I2C

void I2Cstop(void);

Permet d'envoyer la séquence de stop sur le bus I2C

void I2Cwrite_adr(unsigned char data,unsigned char r_w);

Permet d'écrire sur le bus I2C l'adresse du composant I2C sur lequel on veut faire une opération par la suite. On précise ici la nature de cette opération (lecture ou écriture, voir les 2 fonctions ci après) en positionnant la variable r_w.

Exemple :

```
I2Cwrite_adr(ADR_I2C_PCF8574,RD);
```

Indique que l'opération suivante sera une lecture du composant I2C de type PCF8574. Il faut auparavant définir en début de fichier la valeur de ADR_I2C_PCF8574 par

```
#define ADR_I2C_PCF8574 0b01000000
```

void I2Cwrite_data(unsigned char data);

Permet d'envoyer une valeur au composant I2C sélectionné par la fonction I2Cwrite_adr

unsigned char I2Cread(void);

Permet de lire une valeur en provenance du composant I2C sélectionné par la fonction I2Cwrite_adr

7. COMPOSANTS PARTICULIERS

FP10-FP3	Buzzer (hp MINI)	FARNELL	119-1946	0,93€
FP10	IRA E700 (capteur de mouvement)	FARNELL	1006209	2.46€
FP10	Support TO pour IRA E700	FARNELL	177-129	0,53€
FP11	2N3819 (FET)	FARNELL	101-7644	0.21€
FP11-FP22	74ALS874 (8 bascules D)	RADIOSPARES	638-324	2,73€
FP23	Module EZL60 (RS232 <-> Ethernet)	LEXTRONIC	EZL-60	61,04€
FP3	CTN 4.7k 238164073472	FARNELL	1187099	4.94€
FP3	LMC6044 (4 AOP rail to rail)	FARNELL	9494634	3.1€
FP21	Boussole CMP03	SELECTRONIC	90.0660-3	52.5€
FP5	HT9200A-8SOPLF	FARNELL	1420880	0.89€

Prévoir une interface clavier, un module d'affichage LCD et une interface RS232 pour la carte de développement ATMEGA32

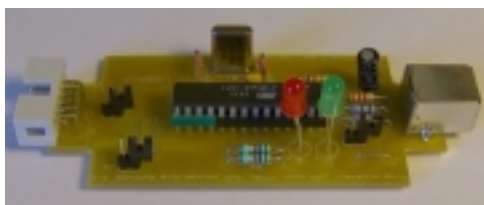
8. LOGICIELS UTILISES

1- CAO circuit imprimé : **PCAD-2004**

2- Programmation ATMEGA :

Ecriture programme C : **AVRStudio4** distribué par ATMEL (gratuit). Pas de limitation de taille de programme

Programmation de l'ATMEGA32 par programmeur USBasp (à réaliser, d'après <http://sti.ac-orleans-tours.fr/spip>) et logiciel **AvrDude** (gratuit). Compatible avec le connecteur HE10 de programmation de la carte de développement.



3- Programmation ATMEGA bis : On a observé sur les ordinateurs élèves à Gray que le programmeur n'était pas correctement reconnu par le bus USB. On a donc opté pour le programmeur ATMEL AVRISP disponible chez FARNELL (1135517 à 28,57€). La différence de prix est négligeable (15€ pour l'USBasp, 29€ pour l'AVRISP). Il faut néanmoins réaliser une carte (simple) d'interface pour la connectique qui n'est pas compatible.

4- Logiciel d'écriture d'une page HTML (sans compilation) : **NotePad++** (gratuit)

5- Logiciel de conversion page HTML en tableau C-source : **WinHex** (version d'évaluation)

6- Logiciel de vérification de communication RS232 : **SuperTerminal** (voir thème 2004 gratuit)

7- Navigateur Internet : **Firefox** (gratuit)