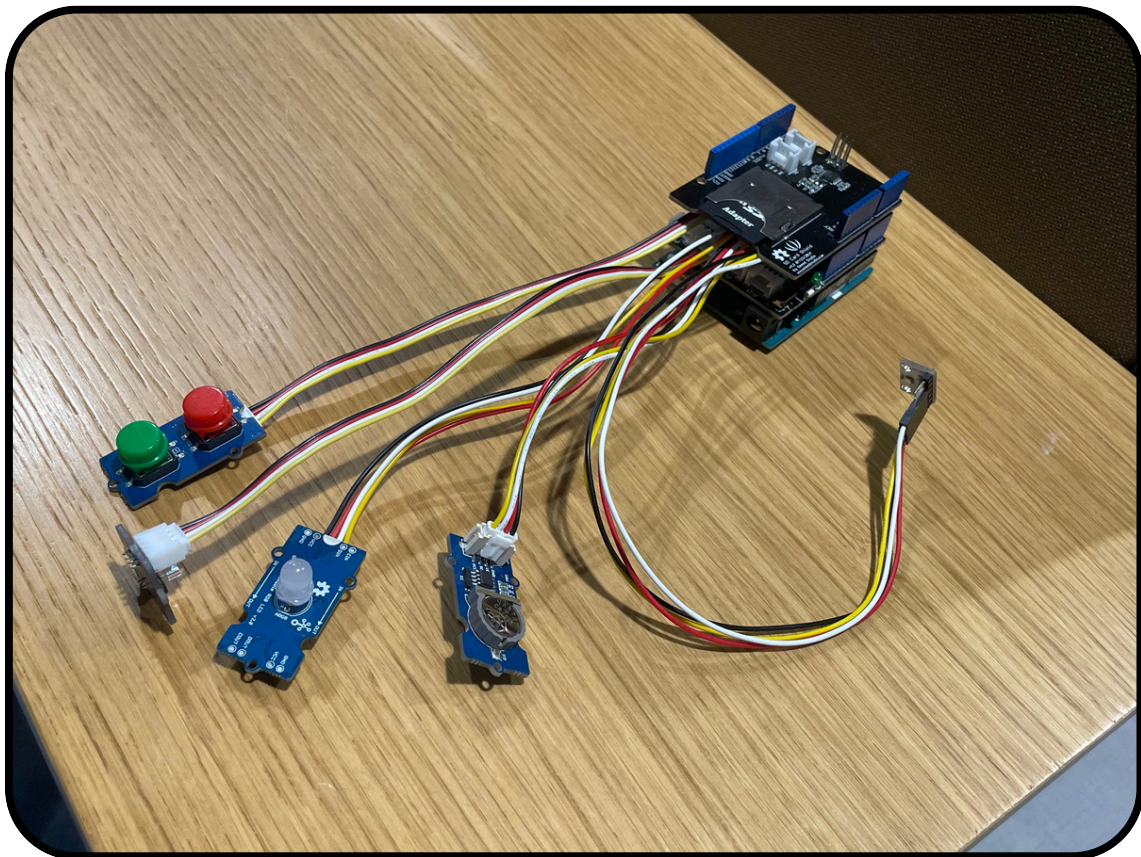


Documentation Technique



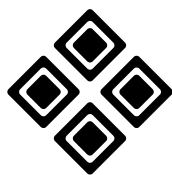
Sommaire :

I Fonctionnement global du système

II Flux d'information (schémas)

III Architecture générale

IV Composants



Fonctionnement global du système

Mode Standard (led): ●

- Automatiquement activé au démarrage.
 - Permet l'acquisition des données
-

Mode Économique(led): ●

- Activation depuis le mode standard :
Appuyer sur le bouton Vert pendant 5 secondes
- Pour sortir du mode économique:
Appuyer sur le bouton Vert pendant 5 secondes
- Multiplie par deux le temps entre chaque mesure

Mode Maintenance(led): ●

- Activation depuis le mode standard :

Appuyer sur le bouton rouge pendant 5 secondes

- Activation depuis le mode économique :

Appuyer sur le bouton vert pendant 5 secondes

- Pour sortir du mode maintenance :

Appuyer sur le bouton rouge pendant 5 secondes

- Ce mode permet de consulter les données via l'interface série et de retirer la carte SD en toute sécurité.
-

Mode Configuration(led): ●













- Pendant le démarrage, maintenir le bouton rouge appuyé
- Pour sortir du mode configuration:

Attendre 30 minutes.

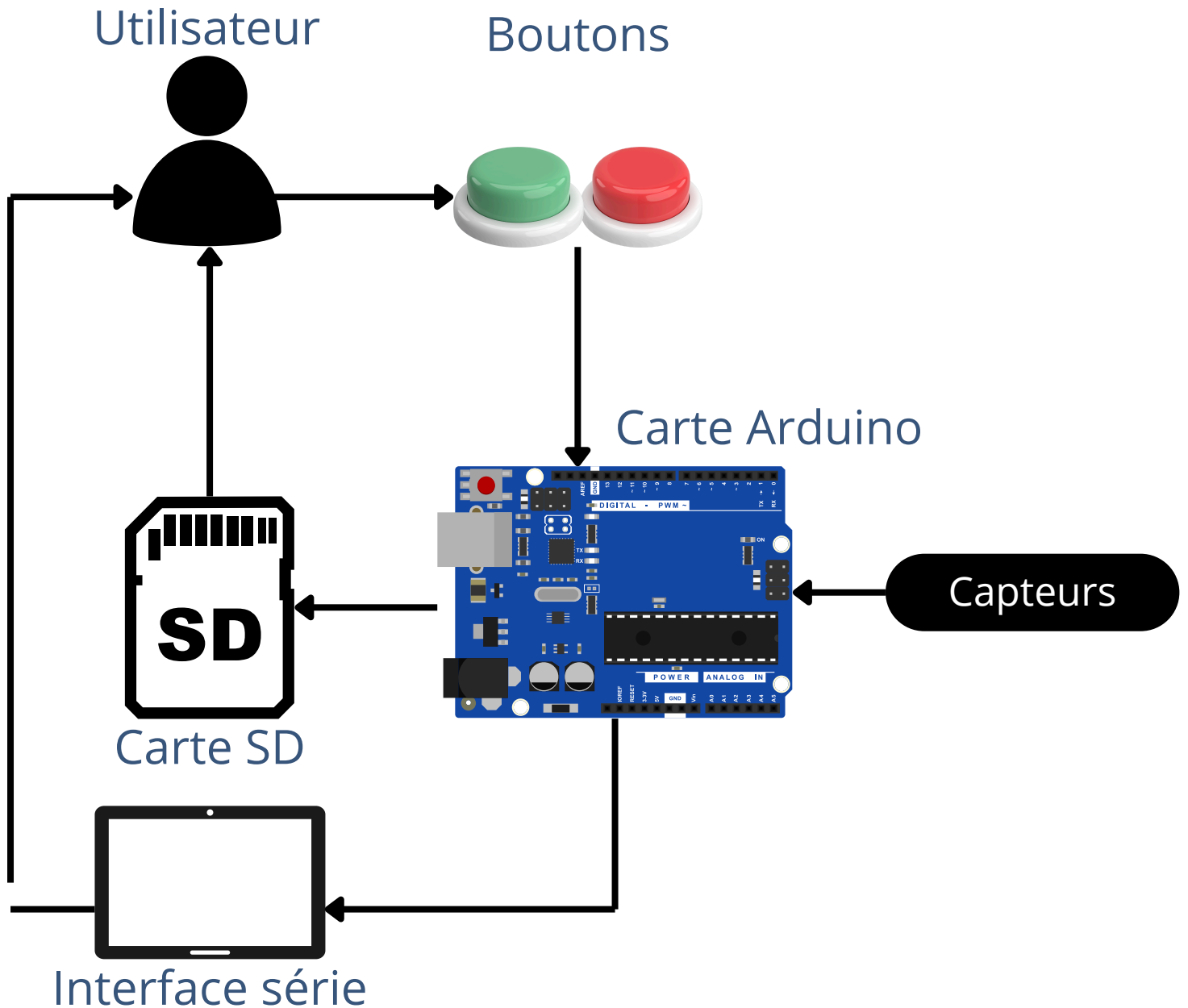
- Permet de configurer les paramètres via l'interface série.

États du système

Système de gestion Erreurs (led clignotement):

-   Carte SD pleine
-   (blanc 2 fois plus long que le rouge)
Erreur d'accès ou d'écriture sur la carte SD
-   Erreur d'accès à l'horloge
-   Erreur d'accès au GPS
-   Erreur d'accès à un capteur
-   (vert 2 fois plus long que le rouge)
Données reçues d'un capteur incohérentes

Flux d'information



Architecture générale

Explication des fonctions

- **Initialisation**

```
void setup {
```

```
  Initialisation LED {}
```

```
  Initialisation bouton vert {}
```

```
  Initialisation bouton rouge {}
```

```
  Bouton vert = digital_read(pinBoutonVert)
```

```
  Bouton rouge = digital_read(pinBoutonRouge)
```

```
  Initialisation interface {}
```

```
  Initialisation horloge {}
```

```
  Initialisation carte SD {}
```

```
  Initialisation GPS {}
```

```
  Initialisation interrupt 1 {Bouton vert appuyé, Mode économique} // Mode économique
```

```
  Initialisation interrupt 2 {Bouton rouge appuyé, Mode maintenance} // Mode Maintenance
```

```
  global eco = 0;
```

```
  global maintenance = 0;
```

```
  global configuration = 0;
```

```
  global LOG_INTERVALL = 10;
```

```
  si bouton rouge pressé :
```

```
    Mode Configuration()
```

```
}
```

- **Boucle Principale**

```
loop {
```

```
  Mode standard()
```

```
}
```

Modes :

Mode configuration() :

```
void Mode configuration() {  
    si 30 min timeout {  
        return  
    } sinon {  
        led_etat("jaune")  
        Etat Systeme()  
        Configuration des paramètres  
        acq_capt_des()  
    }  
}
```

Economie_energie() :

```
void economie_energie() {  
    LOG_INTERVALL = LOG_INTERVALL * 2  
}
```

Changer SD() :

```
void changer SD() {  
    // Permet de changer la carte SD  
}
```

Acq_capt_des() :

```
void acq_capt_des() {  
    // Permet de désactiver l'acquisition  
    des capteurs  
}
```


Fonctionnalités

*Configuration
des
paramètres():*

```
void Configuration des paramètres() {  
    Input Paramètre à accéder  
    si LOG_INTERVALL :  
        Input Nouveau LOG_INTERVALL  
    si TIMEOUT :  
        Input Nouveau TIMEOUT  
    si FILE_MAX_SIZE :  
        Input Nouveau FILE_MAX_SIZE  
    si VERSION :  
        Output Version  
    si CLOCK :  
        input Nouveau CLOCK  
    si DATE :  
        input Nouveau DATE  
    si DAY :  
        input Nouveau DAY  
    si RESET :  
        LOG_INTERVALL= 10  
        TIMEOUT = 30  
        FILE_MAX_SIZE = 4096  
}
```

AccederDonnees() :

```
void accederDonnees() {  
    // Permet d'accéder aux données  
}
```

Récupération Donnees() :

```
void Récupération Donnees() {  
    Lire capteurs  
    Lire horloge  
    Lire GPS  
    Open Carte SD  
    Stocker les données  
    Close Carte SD  
}
```

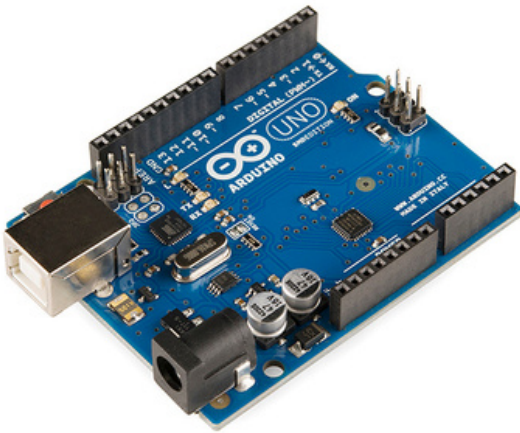
Etat Systeme():

```
void Etat Systeme() {  
    si Erreur RTC {  
        led_etat("rouge_bleu")  
        return  
    }  
    si Erreur GPS {  
        led_etat("rouge_jaune")  
        return  
    }  
    si Erreur GPS données incohérentes {  
        led_etat("rouge_vert1")  
        return  
    }  
    si Données capteur incohérentes {  
        led_etat("rouge_vert2")  
        return  
    }  
    si Carte SD pleine {  
        led_etat("rouge_blanc1")  
        return  
    }  
    si Erreur écriture carte SD {  
        led_etat("rouge_blanc2")  
        return  
    }  
}
```

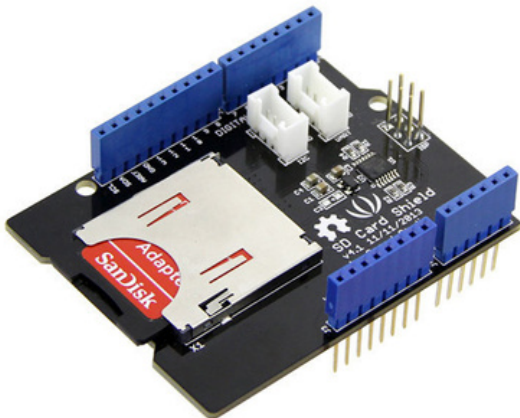
Led_etat () :

```
void led_etat(char etat) {  
    si etats = "bleu"{  
        led -> bleu  
    }  
    sinon si etat == "vert"{  
        led -> vert  
    }  
    sinon si etat == "jaune"{  
        led -> jaune  
    }  
    sinon si etat == "orange"{  
        led -> orange  
    }  
    sinon si etat == "rouge_bleu"{  
        led -> intermittence rouge bleu  
    }  
    sinon si etat == "rouge_jaune"{  
        led -> intermittence rouge jaune  
    }  
    sinon si etat == "rouge_vert1"{  
        led -> intermittence rouge vert 1s  
    }  
    sinon si etat == "rouge_vert2"{  
        led -> intermittence rouge vert 2s  
    }  
    sinon si etat == "rouge_blanche1"{  
        led -> intermittence rouge blanc 1s  
    }  
    sinon si etat == "rouge_blanche2"{  
        led -> intermittence rouge blanc 2s  
    }  
  
}
```

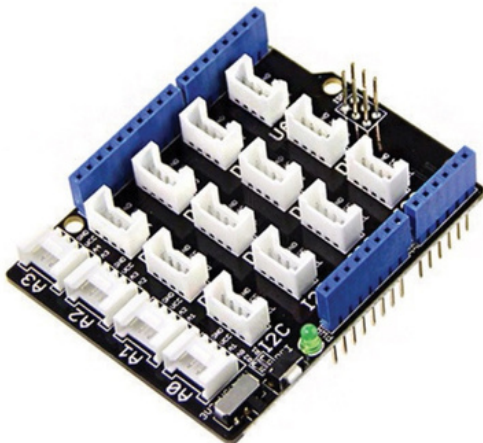
Composants



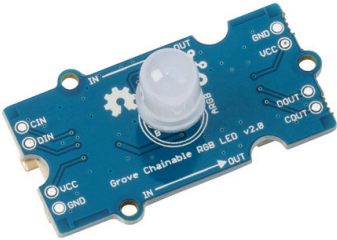
La carte Arduino Uno est basée sur un ATmega328 cadencé à 16 MHz. C'est la plus simple et la plus économique carte à microcontrôleur d'Arduino.



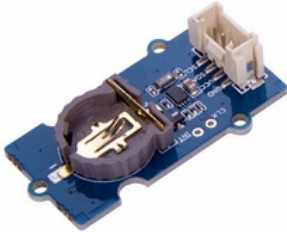
Le shield carte SD est une carte d'interface compatible Arduino permettant d'ajouter un espace de stockage. Il supporte les cartes SD et les cartes micro-SD (via adaptateur inclus).



Le module Grove Base Shield est une carte d'interface permettant de raccorder facilement, rapidement et sans souder les capteurs et les actionneurs Grove de Seeedstudio sur une carte compatible Arduino.



La Led RVB permet d'avoir plusieurs couleurs sur une seul led



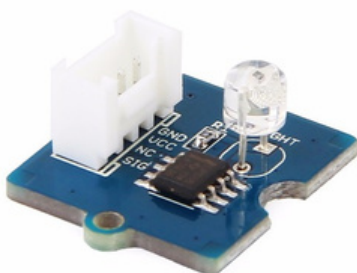
Module RTC de précision compatible Grove basé sur un PCF85063TP à faible consommation. Il donne la date et l'heure au format 12h et 24h, en tenant compte des années bissextiles.



Capteur Grove à double bouton poussoir permettant de contrôler deux canaux de signaux avec un seul module Grove.



Ce capteur est basé sur le circuit BME280 et mesure la température, l'humidité et la pression atmosphérique. Il communique avec un microcontrôleur type Arduino ou compatible via le bus I2C ou SPI.



Le Capteur de Lumière v1.2 Grove permet de mesurer le niveau de lumière