

# Porte du poulailler automatique

v 1.4.1

Généré par Doxygen 1.8.11



# Table des matières

<b>1</b>	<b>Porte-poulailler-automatique</b>	<b>1</b>
<b>2</b>	<b>Index hiérarchique</b>	<b>3</b>
2.1	Hiérarchie des classes . . . . .	3
<b>3</b>	<b>Index des classes</b>	<b>5</b>
3.1	Liste des classes . . . . .	5
<b>4</b>	<b>Index des fichiers</b>	<b>7</b>
4.1	Liste des fichiers . . . . .	7
<b>5</b>	<b>Documentation des classes</b>	<b>9</b>
5.1	Référence de la classe Accus . . . . .	9
5.1.1	Description détaillée . . . . .	10
5.1.2	Documentation des constructeurs et destructeur . . . . .	10
5.1.2.1	Accus(const byte accusPin, const float tensionMiniAccus=4.8, const float rapportConversion=7.5, boolean debug=false) . . . . .	10
5.1.2.2	~Accus() . . . . .	10
5.1.3	Documentation des fonctions membres . . . . .	10
5.1.3.1	accusFaible() . . . . .	10
5.1.3.2	tensionAccus(int valAccus) . . . . .	11
5.1.3.3	tensionAccusCAD() . . . . .	11
5.1.3.4	tensionAccusCADversFloat() . . . . .	12
5.1.4	Documentation des données membres . . . . .	12
5.1.4.1	m_accusPin . . . . .	12
5.1.4.2	m_debug . . . . .	12

5.1.4.3	m_maxCAD	12
5.1.4.4	m_rapportConversion	13
5.1.4.5	m_tensionMiniAccus	13
5.2	Référence de la classe Bouton	13
5.2.1	Description détaillée	15
5.2.2	Documentation des constructeurs et destructeur	16
5.2.2.1	Bouton()	16
5.2.2.2	Bouton(const byte pinBp, const byte pinBoitier, const int debounce=350, const boolean debug=false)	16
5.2.2.3	~Bouton()	16
5.2.3	Documentation des fonctions membres	16
5.2.3.1	testBoitierFerme(const volatile bool &interruptOuvBoi, const bool &boitierOuvert)	16
5.2.3.2	testBoitierOuvert(const volatile bool &interruptOuvBoi, const bool &boitierOuvert)	16
5.2.3.3	testInterruptionBoitier(volatile bool &interruptOuvBoi)	17
5.2.3.4	testInterruptionBp(volatile bool &interruptBp)	17
5.2.3.5	testRelacheBp(volatile bool &interruptBp)	17
5.2.3.6	testToucheBp()	18
5.2.4	Documentation des données membres	18
5.2.4.1	m_debounce	18
5.2.4.2	m_debug	18
5.2.4.3	m_pinBoitier	18
5.2.4.4	m_pinBp	18
5.2.4.5	m_relacheBp	18
5.2.4.6	m_tempoDebounce	18
5.3	Référence de la classe Clavier	19
5.3.1	Description détaillée	21
5.3.2	Documentation des constructeurs et destructeur	21
5.3.2.1	Clavier()	21
5.3.2.2	Clavier(byte nbMenu, const byte pinBp, const byte pinBoitier, const int debounce=350, const boolean debug=false)	21
5.3.2.3	~Clavier()	21

5.3.3	Documentation des fonctions membres . . . . .	21
5.3.3.1	deplacementDansMenu(const &touche, bool &relache, const bool &reglage) . . .	21
5.3.3.2	get_key(int &input) . . . . .	22
5.3.3.3	positionMenu(byte &positionMenu, const byte &toucheClavier) . . . . .	22
5.3.3.4	read_key(const int &sensor) . . . . .	22
5.3.3.5	relacheTouche(const int &touche, bool &relache) . . . . .	23
5.3.3.6	testTouche5(const &touche, bool &relache) . . . . .	23
5.3.4	Documentation des données membres . . . . .	23
5.3.4.1	m_AdckKeyVal . . . . .	23
5.3.4.2	m_debug . . . . .	24
5.3.4.3	m_MenuManuel . . . . .	24
5.3.4.4	m_NumKeys . . . . .	24
5.3.4.5	m_oldKey . . . . .	24
5.4	Référence de la classe Codeur . . . . .	24
5.4.1	Description détaillée . . . . .	25
5.4.2	Documentation des constructeurs et destructeur . . . . .	25
5.4.2.1	Codeur(const byte roueCodeusePin, unsigned int finDeCourseFermeture=250, unsigned int finDeCourseOuverture=150, unsigned int compteRoue↵ Codeuse=150, boolean debug=false) . . . . .	25
5.4.2.2	~Codeur() . . . . .	25
5.4.3	Documentation des fonctions membres . . . . .	26
5.4.3.1	compteurRoueCodeuse(bool ouvFerm) . . . . .	26
5.4.3.2	get_m_compteRoueCodeuse() . . . . .	26
5.4.3.3	get_m_finDeCourseFermeture() . . . . .	26
5.4.3.4	get_m_finDeCourseOuverture() . . . . .	27
5.4.3.5	init() . . . . .	27
5.4.3.6	reglageFinDeCourse(bool ouvFerm, byte touche) . . . . .	27
5.4.3.7	set_m_compteRoueCodeuse(unsigned int compteRoueCodeuse) . . . . .	28
5.4.3.8	set_m_finDeCourseFermeture(unsigned int finDeCourseFermeture) . . . . .	28
5.4.3.9	set_m_finDeCourseOuverture(unsigned int finDeCourseOuverture) . . . . .	28
5.4.3.10	testCompteurRoueCodeuse(byte tolerance) . . . . .	29

5.4.4	Documentation des données membres . . . . .	29
5.4.4.1	m_compteRoueCodeuse . . . . .	29
5.4.4.2	m_debug . . . . .	29
5.4.4.3	m_finDeCourseFermeture . . . . .	29
5.4.4.4	m_finDeCourseMax . . . . .	29
5.4.4.5	m_finDeCourseOuverture . . . . .	29
5.4.4.6	m_interruptRoueCodeuse . . . . .	29
5.4.4.7	m_positionRoueCodeuse . . . . .	30
5.4.4.8	m_roueCodeusePin . . . . .	30
5.5	Référence de la classe HorlogeDS3232 . . . . .	30
5.5.1	Description détaillée . . . . .	32
5.5.2	Documentation des constructeurs et destructeur . . . . .	32
5.5.2.1	HorlogeDS3232() . . . . .	32
5.5.2.2	HorlogeDS3232(const byte adresseMemoireI2C=0x57, const byte m_rtcINT=5, const boolean debug=false) . . . . .	32
5.5.2.3	~HorlogeDS3232() . . . . .	33
5.5.3	Documentation des fonctions membres . . . . .	33
5.5.3.1	bcdToDec(byte val) . . . . .	33
5.5.3.2	calculTemperature(const bool typeTemperature) . . . . .	33
5.5.3.3	choixTypeOuvertureFermeture(bool choixOuvertureFermeture, const byte alarme) . . . . .	34
5.5.3.4	decToBcd(byte val) . . . . .	34
5.5.3.5	get_m_alarm1Hour() . . . . .	34
5.5.3.6	get_m_alarm1Minute() . . . . .	35
5.5.3.7	get_m_alarm1Second() . . . . .	35
5.5.3.8	get_m_alarm2Hour() . . . . .	35
5.5.3.9	get_m_alarm2Minute() . . . . .	36
5.5.3.10	i2c_eeprom_read_byte(unsigned int eeaddress) . . . . .	36
5.5.3.11	i2c_eeprom_write_byte(unsigned int eeaddress, byte data) . . . . .	36
5.5.3.12	init() . . . . .	37
5.5.3.13	lectureHoraireALARM1() . . . . .	37
5.5.3.14	lectureHoraireALARM2() . . . . .	38

5.5.3.15	lectureRegistreEtConversion(byte adresse, byte operationAND=0xFF) . . . . .	38
5.5.3.16	reglageAlarme(const byte touche, const byte alarme, const byte type) . . . . .	39
5.5.3.17	reglageHeure(const byte touche, byte tmDateTime, const byte type) . . . . .	39
5.5.3.18	sauvEepromChoix(unsigned int valeurChoix, const bool matinSoirOuvFerm, const bool lumiereFinDeCourse) . . . . .	40
5.5.3.19	testInterruptRTC(volatile bool &interruptRTC) . . . . .	41
5.5.3.20	testPresenceCarteRTC() . . . . .	41
5.5.4	Documentation des données membres . . . . .	41
5.5.4.1	m_alarm1Hour . . . . .	41
5.5.4.2	m_alarm1Minute . . . . .	41
5.5.4.3	m_alarm1Second . . . . .	41
5.5.4.4	m_alarm2Hour . . . . .	42
5.5.4.5	m_alarm2Minute . . . . .	42
5.5.4.6	m_debug . . . . .	42
5.5.4.7	m_deviceAddress . . . . .	42
5.5.4.8	m_rtcINT . . . . .	42
5.6	Référence de la classe LcdDigoleI2C . . . . .	42
5.6.1	Description détaillée . . . . .	45
5.6.2	Documentation des constructeurs et destructeur . . . . .	45
5.6.2.1	LcdDigoleI2C(TwoWire *, char, byte taille, boolean debug=false) . . . . .	45
5.6.2.2	~LcdDigoleI2C() . . . . .	45
5.6.3	Documentation des fonctions membres . . . . .	45
5.6.3.1	affichageChoix(bool ouverture, bool fermeture, byte ligne) . . . . .	45
5.6.3.2	affichageDateHeure(String jourSemaine, byte jourHeure, byte moisMinute, byte anneeSeconde) . . . . .	46
5.6.3.3	affichageLumFinCourse(unsigned int lum, byte ligne, bool siNonReglable=0) . . . . .	46
5.6.3.4	affichageServo(int pulse, int compteRoueCodeuse, byte ligne) . . . . .	46
5.6.3.5	affichageUneLigne(String chaine) . . . . .	47
5.6.3.6	affichageVoltage(float voltage, String texte, byte ligne) . . . . .	47
5.6.3.7	bonjour(String chaine1, String chaine2) . . . . .	48
5.6.3.8	choixRetroEclairage(bool choix) . . . . .	48

5.6.3.9	cursorPosition(byte decalage, byte ligne, char *texte) . . . . .	49
5.6.3.10	cursorPositionReglages(const byte &touche, bool &relache, bool &reglage, const byte decalageSup, const byte deplacement, const byte decalageInf) . . . . .	49
5.6.3.11	effacementAfficheur() . . . . .	49
5.6.3.12	gestionCurseur(bool curseur) . . . . .	50
5.6.3.13	get_m_decalage() . . . . .	50
5.6.3.14	init() . . . . .	50
5.6.3.15	razLcd() . . . . .	50
5.6.3.16	resetPos(byte ligne) . . . . .	51
5.6.3.17	retroEclairage() . . . . .	51
5.6.3.18	set_m_decalage(byte decalage) . . . . .	51
5.6.3.19	transformation(String chaine, byte dateHeure) . . . . .	51
5.6.4	Documentation des données membres . . . . .	52
5.6.4.1	m_debug . . . . .	52
5.6.4.2	m_decalage . . . . .	52
5.6.4.3	m_deplacement . . . . .	52
5.6.4.4	m_ligne . . . . .	52
5.6.4.5	m_retroEclairage . . . . .	52
5.6.4.6	m_taille . . . . .	52
5.7	Référence de la classe LcdPCF8574 . . . . .	53
5.7.1	Description détaillée . . . . .	55
5.7.2	Documentation des constructeurs et destructeur . . . . .	55
5.7.2.1	LcdPCF8574(char, const byte taille=16, const byte nbLigne=2, boolean debug=false) . . . . .	55
5.7.2.2	~LcdPCF8574() . . . . .	55
5.7.3	Documentation des fonctions membres . . . . .	56
5.7.3.1	affichageChoix(bool ouverture, bool fermeture, byte ligne) . . . . .	56
5.7.3.2	affichageDateHeure(String jourSemaine, byte jourHeure, byte moisMinute, byte anneeSeconde) . . . . .	56
5.7.3.3	affichageLumFinCourse(unsigned int lum, byte ligne, bool siNonReglable=0) . . . . .	56
5.7.3.4	affichageServo(int pulse, int compteRoueCodeuse, byte ligne) . . . . .	57
5.7.3.5	affichageUneLigne(String chaine) . . . . .	57



5.7.3.6	affichageVoltage(float voltage, String texte, byte ligne) . . . . .	58
5.7.3.7	bonjour(String chaine1, String chaine2) . . . . .	58
5.7.3.8	choixRetroEclairage(bool choix) . . . . .	59
5.7.3.9	cursorPosition(byte decalage, byte ligne, char *texte) . . . . .	59
5.7.3.10	cursorPositionReglages(const byte &touche, bool &relache, bool &reglage, const byte decalageSup, const byte deplacement, const byte decalageInf) . . . . .	60
5.7.3.11	effacementAfficheur() . . . . .	60
5.7.3.12	gestionCurseur(bool curseur) . . . . .	61
5.7.3.13	get_m_decalage() . . . . .	61
5.7.3.14	init() . . . . .	61
5.7.3.15	razLcd() . . . . .	61
5.7.3.16	resetPos(byte ligne) . . . . .	61
5.7.3.17	retroEclairage() . . . . .	62
5.7.3.18	set_m_decalage(byte decalage) . . . . .	62
5.7.3.19	transformation(String chaine, byte dateHeure) . . . . .	62
5.7.4	Documentation des données membres . . . . .	62
5.7.4.1	m_debug . . . . .	62
5.7.4.2	m_decalage . . . . .	62
5.7.4.3	m_deplacement . . . . .	62
5.7.4.4	m_ligne . . . . .	62
5.7.4.5	m_retroEclairage . . . . .	63
5.7.4.6	m_taille . . . . .	63
5.8	Référence de la classe Lumiere . . . . .	63
5.8.1	Description détaillée . . . . .	64
5.8.2	Documentation des constructeurs et destructeur . . . . .	64
5.8.2.1	Lumiere(const byte lumierePin, unsigned int lumMatin, unsigned int lumSoir, const byte heureFenetreSoir=17, const float rapportConversion=5, const byte temps↵ Lum=2, boolean debug=false) . . . . .	64
5.8.2.2	~Lumiere() . . . . .	65
5.8.3	Documentation des fonctions membres . . . . .	65
5.8.3.1	declenchementServoLuminosite() . . . . .	65
5.8.3.2	fenetreNonDeclenchement(byte horaire) . . . . .	65

5.8.3.3	get_m_compteurWatchdogLumiere()	66
5.8.3.4	get_m_fermeture()	66
5.8.3.5	get_m_lumMatin()	66
5.8.3.6	get_m_lumSoir()	67
5.8.3.7	get_m_ouverture()	67
5.8.3.8	luminositeCAD()	67
5.8.3.9	nonDeclenchementPositionServo(volatile unsigned int compteRoueCodeuse, unsigned int finDeCourseFermeture, unsigned int finDeCourseOuverture)	68
5.8.3.10	reglageLumiere(bool matinSoir, byte touche)	68
5.8.3.11	set_m_compteurWatchdogLumiere(unsigned int compteurWatchdogLumiere)	69
5.8.3.12	set_m_fermeture(bool fermeture)	69
5.8.3.13	set_m_lumMatin(unsigned int lumMatin)	70
5.8.3.14	set_m_lumSoir(unsigned int lumSoir)	70
5.8.3.15	set_m_ouverture(bool ouverture)	70
5.8.3.16	tensionLuminosite(int valLumiere)	71
5.8.3.17	tensionLuminositeCADversFloat()	71
5.8.3.18	testLuminosite()	71
5.8.4	Documentation des données membres	72
5.8.4.1	m_compteurWatchdogLumiere	72
5.8.4.2	m_debug	72
5.8.4.3	m_fermeture	72
5.8.4.4	m_heureFenetreSoir	72
5.8.4.5	m_incrementation	72
5.8.4.6	m_lumiereMax	72
5.8.4.7	m_lumierePin	72
5.8.4.8	m_lumMatin	72
5.8.4.9	m_lumSoir	73
5.8.4.10	m_maxCAD	73
5.8.4.11	m_ouverture	73
5.8.4.12	m_rapportConversion	73
5.8.4.13	m_tempsLum	73

5.9	Référence de la classe PowerTools . . . . .	73
5.9.1	Description détaillée . . . . .	74
5.9.2	Documentation des constructeurs et destructeur . . . . .	74
5.9.2.1	PowerTools() . . . . .	74
5.9.2.2	PowerTools(const boolean debug=false) . . . . .	74
5.9.2.3	~PowerTools() . . . . .	74
5.9.3	Documentation des fonctions membres . . . . .	75
5.9.3.1	setup_watchdog(int parametreTemps) . . . . .	75
5.9.3.2	setupPower() . . . . .	75
5.9.3.3	tailleChaine(char *chaine) . . . . .	75
5.9.4	Documentation des données membres . . . . .	76
5.9.4.1	m_debug . . . . .	76
5.10	Référence de la classe Radio . . . . .	76
5.10.1	Description détaillée . . . . .	77
5.10.2	Documentation des constructeurs et destructeur . . . . .	77
5.10.2.1	Radio(byte pinEmRadio, byte pinSwitchEmissionRadio, int vitesseTransmission, byte taille, boolean radio=false, boolean debug=false) . . . . .	77
5.10.2.2	~Radio() . . . . .	77
5.10.3	Documentation des fonctions membres . . . . .	78
5.10.3.1	chaineVide() . . . . .	78
5.10.3.2	envoiFloat(float valeur, boolean boitierOuvert=true, char *texte="") . . . . .	78
5.10.3.3	envoiMessage(char *chaine1) . . . . .	79
5.10.3.4	envoiUnsignedInt(unsigned int valeur, boolean boitierOuvert=true, char *texte="") . . . . .	80
5.10.3.5	get_m_radio() . . . . .	80
5.10.3.6	init() . . . . .	81
5.10.3.7	messageRadio(char *chaine1) . . . . .	81
5.10.3.8	messageSansParametre() . . . . .	82
5.10.3.9	raz_m_chaine() . . . . .	82
5.10.3.10	set_m_radio(bool radio) . . . . .	83
5.10.3.11	testSwitchEmissionRadio() . . . . .	83
5.10.4	Documentation des données membres . . . . .	83

5.10.4.1	m_chaine . . . . .	83
5.10.4.2	m_debug . . . . .	83
5.10.4.3	m_pinEmRadio . . . . .	83
5.10.4.4	m_pinSwitchEmissionRadio . . . . .	84
5.10.4.5	m_radio . . . . .	84
5.10.4.6	m_taille . . . . .	84
5.10.4.7	m_vitesseTransmission . . . . .	84
5.11	Référence de la classe ServoMoteur . . . . .	84
5.11.1	Description détaillée . . . . .	87
5.11.2	Documentation des constructeurs et destructeur . . . . .	87
5.11.2.1	ServoMoteur(const byte pinCde, const byte pinRelais, const byte pinSecurite↵ Haute, const int pulseStop=1500, const int pulseOuvFerm=140, const int pulse↵ Réduit=70, boolean debug=false) . . . . .	87
5.11.2.2	~ServoMoteur() . . . . .	87
5.11.3	Documentation des fonctions membres . . . . .	88
5.11.3.1	get_m_ouvFerm() . . . . .	88
5.11.3.2	get_m_pulse() . . . . .	88
5.11.3.3	get_m_servoAction() . . . . .	88
5.11.3.4	init() . . . . .	89
5.11.3.5	modificationVitesse(bool réduit) . . . . .	89
5.11.3.6	relaisHorsTension() . . . . .	90
5.11.3.7	relaisSousTension() . . . . .	90
5.11.3.8	servoHorsTension(unsigned int compteRoueCodeuse, unsigned int finCourse↵ Ouverture) . . . . .	91
5.11.3.9	servoOuvFerm(bool batterieFaible, bool réduit) . . . . .	92
5.11.3.10	servoVitesse(bool réduit) . . . . .	92
5.11.3.11	set_m_ouvFerm(bool ouvFerm) . . . . .	93
5.11.3.12	set_m_pulse(int pulse) . . . . .	94
5.11.3.13	set_m_servoAction(bool servoAction) . . . . .	94
5.11.4	Documentation des données membres . . . . .	94
5.11.4.1	m_debug . . . . .	94
5.11.4.2	m_ouvFerm . . . . .	94
5.11.4.3	m_pinCde . . . . .	94
5.11.4.4	m_pinRelais . . . . .	94
5.11.4.5	m_pinSecuriteHaute . . . . .	94
5.11.4.6	m_pulse . . . . .	94
5.11.4.7	m_pulseOuvFerm . . . . .	95
5.11.4.8	m_pulseRéduit . . . . .	95
5.11.4.9	m_pulseStop . . . . .	95
5.11.4.10	m_servoAction . . . . .	95

<b>6</b>	<b>Documentation des fichiers</b>	<b>97</b>
6.1	Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/carte heuristique porte poulailler.mm . . . . .	97
6.2	Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔affichage_clavier_rtc/AA_fonctions.cpp . . . . .	97
6.3	Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔affichage_clavier_rtc/AA_fonctions.h . . . . .	98
6.3.1	Documentation des fonctions . . . . .	99
6.3.1.1	affichageDemarrage(byte colonne) . . . . .	99
6.3.1.2	affiChoixOuvFerm() . . . . .	100
6.3.1.3	affiFinDeCourseFermeture() . . . . .	100
6.3.1.4	affiFinDeCourseOuverture() . . . . .	101
6.3.1.5	affiLumMatin() . . . . .	102
6.3.1.6	affiLumSoir() . . . . .	102
6.3.1.7	affiPulsePlusCptRoue() . . . . .	103
6.3.1.8	affiTensionBatCdes() . . . . .	104
6.3.1.9	affiTensionBatServo() . . . . .	104
6.3.1.10	choixLumSoir() . . . . .	105
6.3.1.11	choixOuvFerm() . . . . .	105
6.3.1.12	closeTime() . . . . .	106
6.3.1.13	deroulementMenu(byte increment) . . . . .	107
6.3.1.14	displayDate() . . . . .	108
6.3.1.15	displayTime() . . . . .	108
6.3.1.16	eclairageAfficheur() . . . . .	109
6.3.1.17	ecritureDateTime() . . . . .	110
6.3.1.18	enterSleep(void) . . . . .	110
6.3.1.19	fermeturePorte() . . . . .	111
6.3.1.20	ISR(WDT_vect) . . . . .	112
6.3.1.21	lectureClavier() . . . . .	112
6.3.1.22	lumiere() . . . . .	113
6.3.1.23	myInterruptINT0() . . . . .	114
6.3.1.24	myInterruptINT1() . . . . .	115

6.3.1.25	openTime()	115
6.3.1.26	ouverturePorte()	116
6.3.1.27	ouvFermLum()	117
6.3.1.28	read_temp(const boolean typeTemperature)	118
6.3.1.29	regFinDeCourseFermeture()	119
6.3.1.30	regFinDeCourseOuverture()	120
6.3.1.31	reglageDate()	120
6.3.1.32	reglageHeureFermeture()	121
6.3.1.33	reglageHeureOuverture()	121
6.3.1.34	reglageTime()	122
6.3.1.35	routineGestionWatchdog()	123
6.3.1.36	routineInterruptionAlarme2()	123
6.3.1.37	routineInterruptionAlarme1()	124
6.3.1.38	routineInterruptionBp()	125
6.3.1.39	routineTestFermetureBoitier()	125
6.3.1.40	routineTestOuvertureBoitier()	126
6.3.1.41	temporisationAffichage(const int boucleTemps)	126
6.3.1.42	testServo()	127
6.4	Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/Accus.cpp	128
6.5	Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/Accus.h	128
6.6	Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/Bouton.cpp	129
6.7	Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/Bouton.h	130
6.8	Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/Clavier.cpp	131
6.9	Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/Clavier.h	132
6.10	Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/Codeur.cpp	133
6.11	Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/Codeur.h	133

6.12	Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/HorlogeDS3232.cpp . . . . .	134
6.13	Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/HorlogeDS3232.h . . . . .	135
6.13.1	Documentation des macros . . . . .	136
6.13.1.1	DS3231_I2C_ADDRESS . . . . .	136
6.14	Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/LcdDigoleI2C.cpp . . . . .	136
6.15	Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/LcdDigoleI2C.h . . . . .	136
6.15.1	Documentation des macros . . . . .	137
6.15.1.1	_Digole_Serial_I2C_ . . . . .	137
6.16	Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/LcdPCF8574.cpp . . . . .	138
6.17	Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/LcdPCF8574.h . . . . .	138
6.18	Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/Lumiere.cpp . . . . .	139
6.19	Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/Lumiere.h . . . . .	140
6.20	Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/porte_poulailler_affichage_clavier_rtc.ino . . . . .	140
6.20.1	Documentation des macros . . . . .	144
6.20.1.1	LCD_DIGOLE . . . . .	144
6.20.1.2	LED_PIN . . . . .	144
6.20.2	Documentation des fonctions . . . . .	144
6.20.2.1	affichageDemarrage(byte colonne) . . . . .	144
6.20.2.2	affiChoixOuvFerm() . . . . .	145
6.20.2.3	affiFinDeCourseFermeture() . . . . .	145
6.20.2.4	affiFinDeCourseOuverture() . . . . .	146
6.20.2.5	affiLumMatin() . . . . .	147
6.20.2.6	affiLumSoir() . . . . .	147
6.20.2.7	affiPulsePlusCptRoue() . . . . .	148
6.20.2.8	affiTensionBatCdes() . . . . .	149

6.20.2.9	affiTensionBatServo()	149
6.20.2.10	choixLumMatin()	150
6.20.2.11	choixLumSoir()	150
6.20.2.12	choixOuvFerm()	151
6.20.2.13	closeTime()	152
6.20.2.14	deroulementMenu(byte increment)	153
6.20.2.15	displayDate()	154
6.20.2.16	displayTime()	154
6.20.2.17	eclairageAfficheur()	155
6.20.2.18	ecritureDateTime()	156
6.20.2.19	enterSleep(void)	156
6.20.2.20	fermeturePorte()	157
6.20.2.21	ISR(WDT_vect)	157
6.20.2.22	lectureClavier()	158
6.20.2.23	loop()	159
6.20.2.24	lumiere()	159
6.20.2.25	myInterruptINT0()	160
6.20.2.26	myInterruptINT1()	161
6.20.2.27	openTime()	161
6.20.2.28	ouverturePorte()	162
6.20.2.29	ouvFermLum()	163
6.20.2.30	read_temp(const boolean typeTemperature)	164
6.20.2.31	regFinDeCourseFermeture()	165
6.20.2.32	regFinDeCourseOuverture()	166
6.20.2.33	reglageDate()	166
6.20.2.34	reglageHeureFermeture()	167
6.20.2.35	reglageHeureOuverture()	167
6.20.2.36	reglageTime()	168
6.20.2.37	routineGestionWatchdog()	169
6.20.2.38	routineInterruptionAlarme2()	169



6.20.2.39	routineInterruptionAlarme1()	170
6.20.2.40	routineInterruptionBp()	171
6.20.2.41	routineTestFermetureBoitier()	171
6.20.2.42	routineTestOuvertureBoitier()	172
6.20.2.43	setup()	172
6.20.2.44	temporisationAffichage(const int boucleTemps)	173
6.20.2.45	testServo()	174
6.20.3	Documentation des variables	175
6.20.3.1	accusCde	175
6.20.3.2	accusPinCde	175
6.20.3.3	accusPinServo	175
6.20.3.4	accusServo	175
6.20.3.5	adresseBoitier24C32	175
6.20.3.6	alarm_1	175
6.20.3.7	alarm_2	175
6.20.3.8	annee	176
6.20.3.9	batterieFaible	176
6.20.3.10	boitierOuvert	176
6.20.3.11	boucleLumiere	176
6.20.3.12	bouclesWatchdog	176
6.20.3.13	boucleTemps	176
6.20.3.14	clav	176
6.20.3.15	codOpt	176
6.20.3.16	colonnes	176
6.20.3.17	compteRoueCodeuse	176
6.20.3.18	conversion	176
6.20.3.19	debounce	177
6.20.3.20	debug	177
6.20.3.21	emissionRadio	177
6.20.3.22	f_wdt	177

6.20.3.23 finDeCourseFermeture . . . . .	177
6.20.3.24 finDeCourseOuverture . . . . .	177
6.20.3.25 heure . . . . .	177
6.20.3.26 heureFenetreSoir . . . . .	177
6.20.3.27 incrementation . . . . .	177
6.20.3.28 interruptBp . . . . .	178
6.20.3.29 interruptOuvBoi . . . . .	178
6.20.3.30 interruptRTC . . . . .	178
6.20.3.31 jour . . . . .	178
6.20.3.32 jourSemaine . . . . .	178
6.20.3.33 LcdCursor . . . . .	178
6.20.3.34 lum . . . . .	178
6.20.3.35 lumierePin . . . . .	178
6.20.3.36 lumMatin . . . . .	178
6.20.3.37 lumSoir . . . . .	178
6.20.3.38 memoireLibre . . . . .	179
6.20.3.39 menuChoix . . . . .	179
6.20.3.40 menuDate . . . . .	179
6.20.3.41 menuFermeture . . . . .	179
6.20.3.42 menuFinDeCourseFermeture . . . . .	179
6.20.3.43 menuFinDeCourseOuverture . . . . .	179
6.20.3.44 menuHeure . . . . .	179
6.20.3.45 menuLumiere . . . . .	179
6.20.3.46 menuLumiereMatin . . . . .	179
6.20.3.47 menuLumiereSoir . . . . .	179
6.20.3.48 menuManuel . . . . .	180
6.20.3.49 menuOuverture . . . . .	180
6.20.3.50 menuTemperature . . . . .	180
6.20.3.51 menuTensionBatCdes . . . . .	180
6.20.3.52 menuTensionBatServo . . . . .	180

6.20.3.53 minutes . . . . .	180
6.20.3.54 mois . . . . .	180
6.20.3.55 monServo . . . . .	180
6.20.3.56 oldkey . . . . .	180
6.20.3.57 pinBoitier . . . . .	180
6.20.3.58 pinBp . . . . .	180
6.20.3.59 pinEmRadio . . . . .	181
6.20.3.60 pinSwitchEmissionRadio . . . . .	181
6.20.3.61 PROGMEM . . . . .	181
6.20.3.62 pulseStop . . . . .	181
6.20.3.63 radio . . . . .	181
6.20.3.64 rapportConversion . . . . .	181
6.20.3.65 reduit . . . . .	181
6.20.3.66 reglage . . . . .	181
6.20.3.67 relache . . . . .	181
6.20.3.68 roueCodeuse . . . . .	181
6.20.3.69 rtc . . . . .	182
6.20.3.70 rtcINT . . . . .	182
6.20.3.71 secondes . . . . .	182
6.20.3.72 securiteHaute . . . . .	182
6.20.3.73 sensorClavier . . . . .	182
6.20.3.74 servoCde . . . . .	182
6.20.3.75 servoPin . . . . .	182
6.20.3.76 temps . . . . .	182
6.20.3.77 tempsWatchdog . . . . .	182
6.20.3.78 tensionMiniAccus . . . . .	182
6.20.3.79 testServoMoteur . . . . .	183
6.20.3.80 tm . . . . .	183
6.20.3.81 tools . . . . .	183
6.20.3.82 touche . . . . .	183

6.20.3.83 typeTemperature . . . . .	183
6.20.3.84 vitesseTransmission . . . . .	183
6.20.3.85 Wire . . . . .	183
6.21 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/PowerTools.cpp . . . . .	183
6.22 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/PowerTools.h . . . . .	184
6.23 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/Radio.cpp . . . . .	184
6.24 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/Radio.h . . . . .	185
6.25 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/ServoMoteur.cpp . . . . .	186
6.26 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_↔ affichage_clavier_rtc/ServoMoteur.h . . . . .	187
6.27 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/README.md . . .	187
<b>Index</b>	<b>189</b>

## Chapitre 1

# Porte-poulailler-automatique

Automatisation de la porte de mon poulailler. Déclenchement par la lumière ou l'heure.



## Chapitre 2

# Index hiérarchique

### 2.1 Hiérarchie des classes

Cette liste d'héritage est classée approximativement par ordre alphabétique :

Accus . . . . .	9
Bouton . . . . .	13
Clavier . . . . .	19
Codeur . . . . .	24
DigoleSerialDisp	
LcdDigoleI2C . . . . .	42
DS3232RTC	
HorlogeDS3232 . . . . .	30
LiquidCrystal_I2C	
LcdPCF8574 . . . . .	53
Lumiere . . . . .	63
PowerTools . . . . .	73
Radio . . . . .	76
ServoTimer2	
ServoMoteur . . . . .	84





## Chapitre 3

# Index des classes

### 3.1 Liste des classes

Liste des classes, structures, unions et interfaces avec une brève description :

<b>Accus</b>		
	Gestion des blocs accumulateurs NIMH de 4,8v . . . . .	??
<b>Bouton</b>	. . . . .	??
<b>Clavier</b>	. . . . .	??
<b>Codeur</b>	. . . . .	??
<b>HorlogeDS3232</b>	. . . . .	??
<b>LcdDigoleI2C</b>	. . . . .	??
<b>LcdPCF8574</b>	. . . . .	??
<b>Lumiere</b>	. . . . .	??
<b>PowerTools</b>	. . . . .	??
<b>Radio</b>	. . . . .	??
<b>ServoMoteur</b>	. . . . .	??



## Chapitre 4

# Index des fichiers

### 4.1 Liste des fichiers

Liste de tous les fichiers avec une brève description :

/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/ <b>carte heuristique porte poulailler.mm</b>	...	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>AA_↵</b>		
<b>fonctions.cpp</b>	...	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>AA_↵</b>		
<b>fonctions.h</b>	...	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>Accus.↵</b>		
<b>cpp</b>	...	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>Accus.↵</b>		
<b>h</b>	...	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>Bouton.↵</b>		
<b>cpp</b>	...	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>Bouton.↵</b>		
<b>h</b>	...	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>Clavier.↵</b>		
<b>cpp</b>	...	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>Clavier.↵</b>		
<b>h</b>	...	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>Codeur.↵</b>		
<b>cpp</b>	...	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>Codeur.↵</b>		
<b>h</b>	...	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>Horloge↵</b>		
<b>DS3232.cpp</b>	...	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>Horloge↵</b>		
<b>DS3232.h</b>	...	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>Lcd↵</b>		
<b>Digolel2C.cpp</b>	...	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>Lcd↵</b>		
<b>Digolel2C.h</b>	...	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>LcdP↵</b>		
<b>CF8574.cpp</b>	...	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>LcdP↵</b>		
<b>CF8574.h</b>	...	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>Lumiere.↵</b>		
<b>cpp</b>	...	??

/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>Lumiere.</b> ↵	
<b>h</b> . . . . .	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>porte</b> ↵	
<b>_poulailler_affichage_clavier_rtc.ino</b> . . . . .	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>Power</b> ↵	
<b>Tools.cpp</b> . . . . .	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>Power</b> ↵	
<b>Tools.h</b> . . . . .	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>Radio.</b> ↵	
<b>cpp</b> . . . . .	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>Radio.</b> ↵	
<b>h</b> . . . . .	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>Servo</b> ↵	
<b>Moteur.cpp</b> . . . . .	??
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/ <b>Servo</b> ↵	
<b>Moteur.h</b> . . . . .	??

## Chapitre 5

# Documentation des classes

### 5.1 Référence de la classe Accus

Gestion des blocs accumulateurs NIMH de 4,8v.

```
#include <Accus.h>
```

Graphe de collaboration de Accus :

Accus
# m_debug # m_accusPin # m_rapportConversion # m_maxCAD # m_tensionMiniAccus
+ Accus() + ~Accus() + accusFaible() + tensionAccusCAD() + tensionAccus() + tensionAccusCADversFloat()

#### Fonctions membres publiques

- **Accus** (const byte accusPin, const float **tensionMiniAccus**=4.8, const float **rapportConversion**=7.5, boolean **debug**=false)
- **~Accus** ()
- bool **accusFaible** ()  
*teste si la tension est faible*
- int **tensionAccusCAD** ()  
*lecture tension batterie CAD*
- float **tensionAccus** (int valAccus)  
*conversion CAD vers tension batterie*
- float **tensionAccusCADversFloat** ()  
*lecture et conversion vers tension batterie en float*

## Attributs protégés

- const boolean **m\_debug**
- const byte **m\_accusPin**
- const float **m\_rapportConversion**
- const int **m\_maxCAD**
- const float **m\_tensionMiniAccus**

### 5.1.1 Description détaillée

Gestion des blocs accumulateurs NIMH de 4,8v.

Suivi de la tension et la valeur minimum

#### Auteur

Zephyr5028

#### Date

hiver 2017 Prototype de la classe **Accus** (p. ??)

Définition à la ligne 16 du fichier Accus.h.

### 5.1.2 Documentation des constructeurs et destructeur

**5.1.2.1** **Accus** : **Accus** ( const byte *accusPin*, const float *tensionMiniAccus* = 4 . 8, const float *rapportConversion* = 7 . 5, boolean *debug* = false )

**Accus.cpp** (p. ??) définitions de la classe **Accus** (p. ??) bloc accus NIMH

Définition à la ligne 9 du fichier Accus.cpp.

**5.1.2.2** **Accus** : **~Accus** ( )

Définition à la ligne 14 du fichier Accus.cpp.

### 5.1.3 Documentation des fonctions membres

**5.1.3.1** bool **Accus** : **accusFaible** ( )

teste si la tension est faible

Lecture et conversion de la valeur mesurée sur un pont de résistances en parallèle de l'accu

#### Paramètres

<i>m_accusPin</i>	numéro de l'entrée analogique
-------------------	-------------------------------

## Renvoi

batterieFaible si  $< 4,8v$

conversion tension mini en pas du CAD

read the input on analog pin tension batterie

si la batterie est faible  $< 4,8v$  (654)

Définition à la ligne 24 du fichier Accus.cpp.

Voici le graphe des appelants de cette fonction :

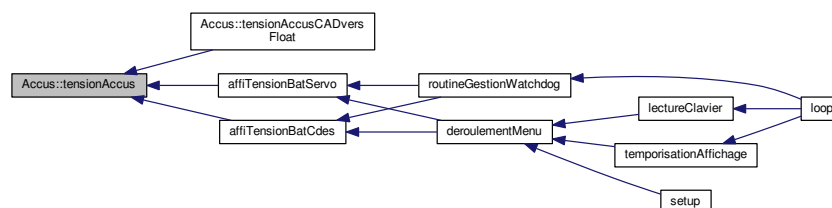


## 5.1.3.2 float Accus : :tensionAccus ( int valAccus )

----- conversion CAD vers tension batterie -----

Définition à la ligne 42 du fichier Accus.cpp.

Voici le graphe des appelants de cette fonction :

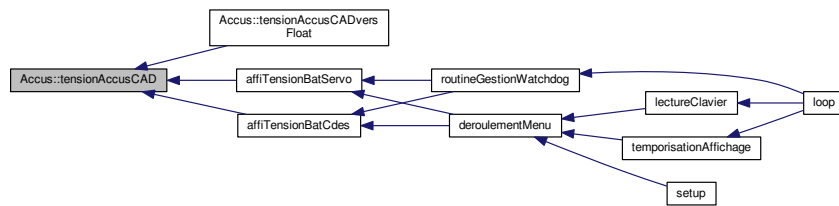


## 5.1.3.3 int Accus : :tensionAccusCAD ( )

----- lecture tension batterie CAD-----

Définition à la ligne 36 du fichier Accus.cpp.

Voici le graphe des appelants de cette fonction :

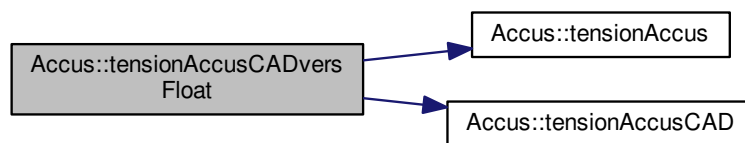


#### 5.1.3.4 float Accus : :tensionAccusCADversFloat ( )

---lecture et conversion vers tension batterie en float---

Définition à la ligne 49 du fichier Accus.cpp.

Voici le graphe d'appel pour cette fonction :



### 5.1.4 Documentation des données membres

#### 5.1.4.1 const byte Accus : :m\_accusPin [protected]

Définition à la ligne 33 du fichier Accus.h.

#### 5.1.4.2 const boolean Accus : :m\_debug [protected]

Définition à la ligne 32 du fichier Accus.h.

#### 5.1.4.3 const int Accus : :m\_maxCAD [protected]

Définition à la ligne 36 du fichier Accus.h.



5.1.4.4 `const float Accus : :m_rapportConversion` [protected]

Définition à la ligne 35 du fichier Accus.h.

5.1.4.5 `const float Accus : :m_tensionMiniAccus` [protected]

Définition à la ligne 37 du fichier Accus.h.

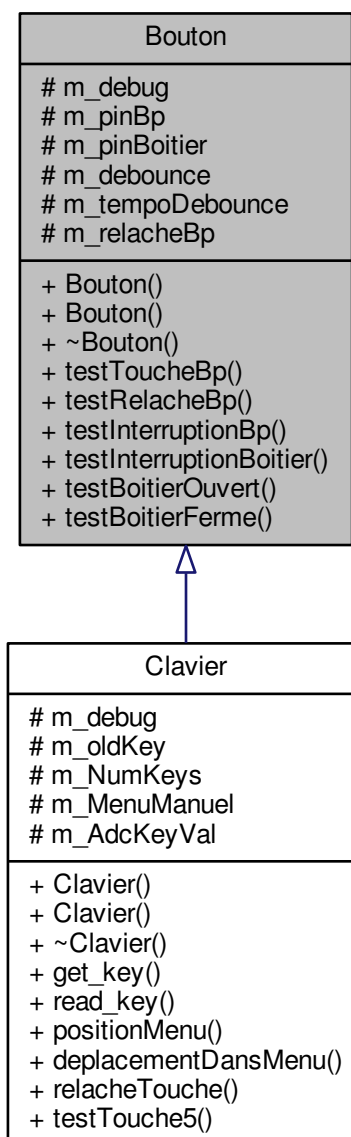
La documentation de cette classe a été générée à partir des fichiers suivants :

- /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/**Accus.h**
- /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/**Accus.cpp**

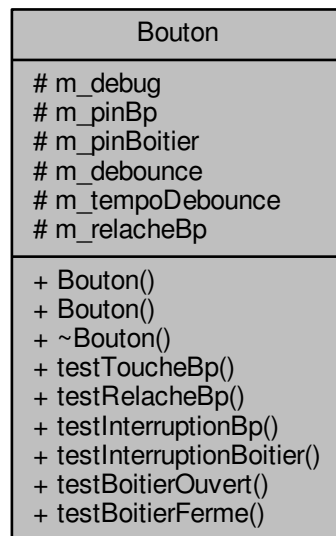
## 5.2 Référence de la classe Bouton

```
#include <Bouton.h>
```

Graphe d'héritage de Bouton :



Graphe de collaboration de Bouton :



### Fonctions membres publiques

```
— Bouton ()
— Bouton (const byte pinBp, const byte pinBoitier, const int debounce=350, const boolean debug=false)
— ~Bouton ()
— bool testToucheBp ()
    ———test touche Bp———
— void testRelacheBp (volatile bool &interruptBp)
    ———test relache Bp———
— void testInterruptionBp (volatile bool &interruptBp)
    ———test IT Bp———
— void testInterruptionBoitier (volatile bool &interruptOuvBoi)
    ———test IT ouverture boitier———
— bool testBoitierOuvert (const volatile bool &interruptOuvBoi, const bool &boitierOuvert)
    ———test boitier ouvert———
— bool testBoitierFerme (const volatile bool &interruptOuvBoi, const bool &boitierOuvert)
    ———test boitier ferme———
```

### Attributs protégés

```
— const boolean m_debug
— const byte m_pinBp
— const byte m_pinBoitier
— const int m_debounce
— int m_tempoDebounce
— bool m_relacheBp
```

#### 5.2.1 Description détaillée

**Bouton.h** (p. ??) prototype de la **Bouton** (p. ??)

Définition à la ligne 10 du fichier Bouton.h.

## 5.2.2 Documentation des constructeurs et destructeur

### 5.2.2.1 Bouton : :Bouton ( )

**Bouton.cpp** (p. ??) définitions de la classe **Bouton** (p. ??)

Définition à la ligne 7 du fichier Bouton.cpp.

```
5.2.2.2 Bouton : :Bouton ( const byte pinBp, const byte pinBoitier, const int debounce = 350, const boolean debug =
false )
```

Définition à la ligne 12 du fichier Bouton.cpp.

### 5.2.2.3 Bouton : :~Bouton ( )

Définition à la ligne 18 du fichier Bouton.cpp.

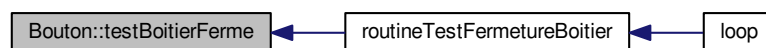
## 5.2.3 Documentation des fonctions membres

### 5.2.3.1 bool Bouton : :testBoitierFerme ( const volatile bool & *interruptOuvBoi*, const bool & *boitierOuvert* )

----test boitier ferme----

Définition à la ligne 66 du fichier Bouton.cpp.

Voici le graphe des appelants de cette fonction :

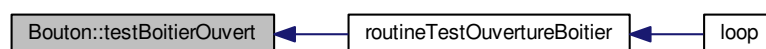


### 5.2.3.2 bool Bouton : :testBoitierOuvert ( const volatile bool & *interruptOuvBoi*, const bool & *boitierOuvert* )

----test boitier ouvert----

Définition à la ligne 57 du fichier Bouton.cpp.

Voici le graphe des appelants de cette fonction :

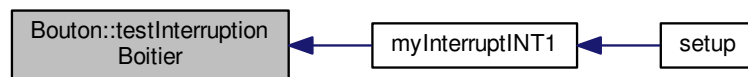


### 5.2.3.3 void Bouton : :testInterruptionBoitier ( volatile bool & interruptOuvBoi )

---test IT ouverture boitier---

Définition à la ligne 50 du fichier Bouton.cpp.

Voici le graphe des appelants de cette fonction :



### 5.2.3.4 void Bouton : :testInterruptionBp ( volatile bool & interruptBp )

---test IT Bp---

Définition à la ligne 42 du fichier Bouton.cpp.

Voici le graphe des appelants de cette fonction :

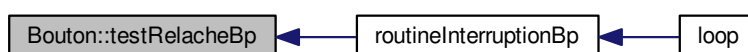


### 5.2.3.5 void Bouton : :testRelacheBp ( volatile bool & interruptBp )

---test relache Bp---

Définition à la ligne 33 du fichier Bouton.cpp.

Voici le graphe des appelants de cette fonction :

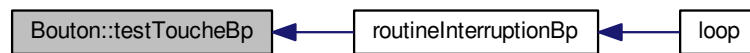


### 5.2.3.6 `bool Bouton : :testToucheBp ( )`

---test touche Bp---

Définition à la ligne 23 du fichier Bouton.cpp.

Voici le graphe des appelants de cette fonction :



## 5.2.4 Documentation des données membres

### 5.2.4.1 `const int Bouton : :m_debounce [protected]`

Définition à la ligne 30 du fichier Bouton.h.

### 5.2.4.2 `const boolean Bouton : :m_debug [protected]`

Définition à la ligne 27 du fichier Bouton.h.

### 5.2.4.3 `const byte Bouton : :m_pinBoitier [protected]`

Définition à la ligne 29 du fichier Bouton.h.

### 5.2.4.4 `const byte Bouton : :m_pinBp [protected]`

Définition à la ligne 28 du fichier Bouton.h.

### 5.2.4.5 `bool Bouton : :m_relacheBp [protected]`

Définition à la ligne 32 du fichier Bouton.h.

### 5.2.4.6 `int Bouton : :m_tempoDebounce [protected]`

Définition à la ligne 31 du fichier Bouton.h.

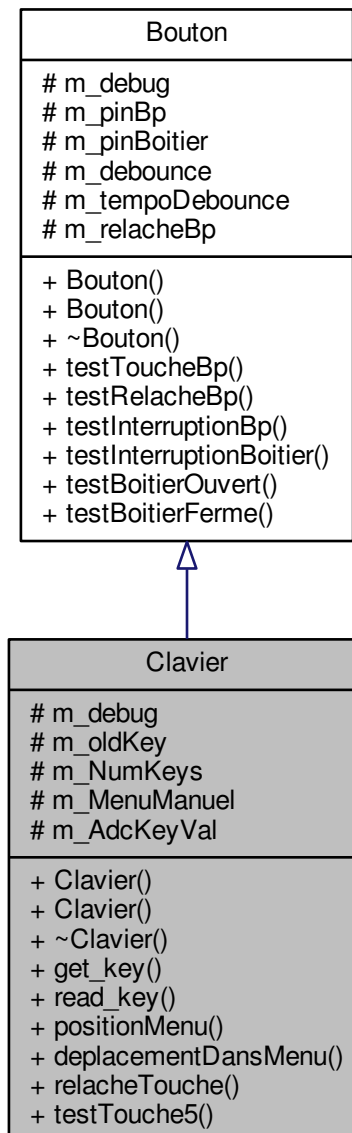
La documentation de cette classe a été générée à partir des fichiers suivants :

- /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/**Bouton.h**↔
- /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/**Bouton.cpp**↔

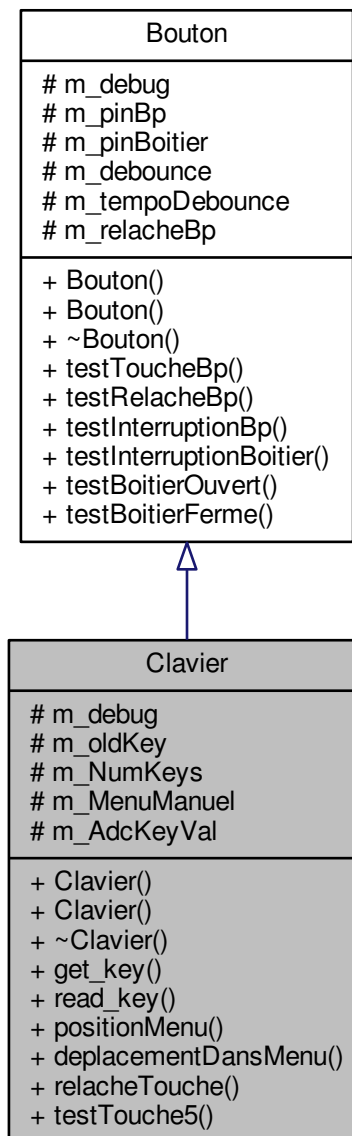
## 5.3 Référence de la classe Clavier

```
#include <Clavier.h>
```

Graphe d'héritage de Clavier :



Graphe de collaboration de Clavier :



## Fonctions membres publiques

- **Clavier** ()
- **Clavier** (byte nbMenu, const byte **pinBp**, const byte **pinBoitier**, const int **debounce**=350, const boolean **debug**=false)
- **~Clavier** ()
- int **get\_key** (int &input)
  - routine *get\_key* : Convert ADC value to key number-----
- int **read\_key** (const int &sensor)
  - routine *read\_key* : lecture du clavier-----
- void **positionMenu** (byte &positionMenu, const byte &toucheClavier)
  - position du menu pour l'affichage - touches 2 et 3 -----



- bool **deplacementDansMenu** (const &**touche**, bool &**relache**, const bool &**reglage**)  
     *---deplacement dans le menu---*
- void **relacheTouche** (const int &**touche**, bool &**relache**)  
     *---relache touche---*
- bool **testTouche5** (const &**touche**, bool &**relache**)  
     *---test touche 5---*

### Attributs protégés

- const boolean **m\_debug**
- const byte **m\_oldKey**
- const byte **m\_NumKeys**
- byte **m\_MenuManuel**
- int **m\_AdckKeyVal** [5]

### 5.3.1 Description détaillée

**Clavier.h** (p. ??) prototype de la classe **Clavier** (p. ??)

Définition à la ligne 11 du fichier Clavier.h.

### 5.3.2 Documentation des constructeurs et destructeur

#### 5.3.2.1 Clavier : :Clavier ( )

**Clavier.cpp** (p. ??) définitions de la classe **Clavier** (p. ??)

Définition à la ligne 7 du fichier Clavier.cpp.

#### 5.3.2.2 Clavier : :Clavier ( byte nbMenu, const byte pinBp, const byte pinBoitier, const int debounce = 350, const boolean debug = false )

Définition à la ligne 13 du fichier Clavier.cpp.

#### 5.3.2.3 Clavier : :~Clavier ( )

Définition à la ligne 20 du fichier Clavier.cpp.

### 5.3.3 Documentation des fonctions membres

#### 5.3.3.1 bool Clavier : :deplacementDansMenu ( const & touche, bool & relache, const bool & reglage )

*---deplacement dans le menu---*

Définition à la ligne 79 du fichier Clavier.cpp.

Voici le graphe des appelants de cette fonction :

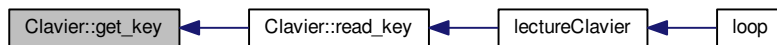


### 5.3.3.2 int Clavier : :get\_key ( int & input )

-----routine get\_key : Convert ADC value to key number-----

Définition à la ligne 25 du fichier Clavier.cpp.

Voici le graphe des appelants de cette fonction :



### 5.3.3.3 void Clavier : :positionMenu ( byte & positionMenu, const byte & toucheClavier )

-----position du menu pour l'affichage - touches 2 et 3 -----

Définition à la ligne 54 du fichier Clavier.cpp.

Voici le graphe des appelants de cette fonction :



### 5.3.3.4 int Clavier : :read\_key ( const int & sensor )

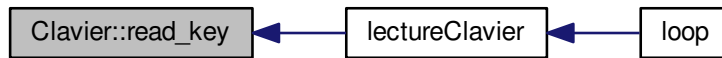
-----routine read\_key : lecture du clavier-----

Définition à la ligne 39 du fichier Clavier.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 5.3.3.5 void Clavier : :relacheTouche ( const int & *touche*, bool & *relache* )

---relache touche---

Définition à la ligne 72 du fichier Clavier.cpp.

Voici le graphe des appelants de cette fonction :

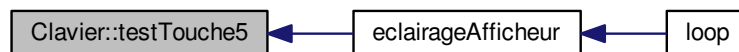


#### 5.3.3.6 bool Clavier : :testTouche5 ( const & *touche*, bool & *relache* )

---test touche 5---

Définition à la ligne 89 du fichier Clavier.cpp.

Voici le graphe des appelants de cette fonction :



### 5.3.4 Documentation des données membres

#### 5.3.4.1 int Clavier : :m\_AdckKeyVal[5] [protected]

Définition à la ligne 33 du fichier Clavier.h.

#### 5.3.4.2 `const boolean Clavier : :m_debug` [protected]

Définition à la ligne 29 du fichier Clavier.h.

#### 5.3.4.3 `byte Clavier : :m_MenuManuel` [protected]

Définition à la ligne 32 du fichier Clavier.h.

#### 5.3.4.4 `const byte Clavier : :m_NumKeys` [protected]

Définition à la ligne 31 du fichier Clavier.h.

#### 5.3.4.5 `const byte Clavier : :m_oldKey` [protected]

Définition à la ligne 30 du fichier Clavier.h.

La documentation de cette classe a été générée à partir des fichiers suivants :

- /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/**Clavier.h**
- /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/**Clavier.cpp**

## 5.4 Référence de la classe Codeur

```
#include <Codeur.h>
```

Grphe de collaboration de Codeur :

Codeur
<pre># m_roueCodeusePin # m_finDeCourseMax # m_debug # m_compteRoueCodeuse # m_interruptRoueCodeuse # m_positionRoueCodeuse # m_finDeCourseFermeture # m_finDeCourseOuverture</pre>
<pre>+ Codeur() + ~Codeur() + init() + testCompteurRoueCodeuse() + reglageFinDeCourse() + compteurRoueCodeuse() + get_m_compteRoueCodeuse() + set_m_compteRoueCodeuse() + get_m_finDeCourseFermeture() + set_m_finDeCourseFermeture() + get_m_finDeCourseOuverture() + set_m_finDeCourseOuverture()</pre>

## Fonctions membres publiques

- **Codeur** (const byte *roueCodeusePin*, unsigned int **finDeCourseFermeture**=250, unsigned int **finDeCourseOuverture**=150, unsigned int **compteRoueCodeuse**=150, boolean **debug**=false)
- **~Codeur** ()
- void **init** ()
- *---initialisation---*
- byte **testCompteurRoueCodeuse** (byte *tolerance*)
- *---test du compteur roue codeuse pour affichage ouvert ou ferme---*
- unsigned int **reglageFinDeCourse** (bool *ouvFerm*, byte **touche**)
- *---reglage de la fin de course---*
- void **compteurRoueCodeuse** (bool *ouvFerm*)
- *---compteur roue codeuse---*
- unsigned int **get\_m\_compteRoueCodeuse** ()
- void **set\_m\_compteRoueCodeuse** (unsigned int *compteRoueCodeuse*)
- unsigned int **get\_m\_finDeCourseFermeture** ()
- void **set\_m\_finDeCourseFermeture** (unsigned int *finDeCourseFermeture*)
- unsigned int **get\_m\_finDeCourseOuverture** ()
- void **set\_m\_finDeCourseOuverture** (unsigned int *finDeCourseOuverture*)

## Attributs protégés

- const byte **m\_roueCodeusePin**
- const unsigned int **m\_finDeCourseMax**
- const boolean **m\_debug**
- volatile unsigned int **m\_compteRoueCodeuse**
- volatile boolean **m\_interruptRoueCodeuse**
- volatile boolean **m\_positionRoueCodeuse**
- unsigned int **m\_finDeCourseFermeture**
- unsigned int **m\_finDeCourseOuverture**

## 5.4.1 Description détaillée

**Codeur.h** (p. ??) prototype de la classe **Codeur** (p. ??) codeur optique de 20 pas

Définition à la ligne 11 du fichier Codeur.h.

## 5.4.2 Documentation des constructeurs et destructeur

5.4.2.1 **Codeur** : **Codeur** ( const byte *roueCodeusePin*, unsigned int *finDeCourseFermeture* = 250, unsigned int *finDeCourseOuverture* = 150, unsigned int *compteRoueCodeuse* = 150, boolean *debug* = false )

**Codeur.cpp** (p. ??) définitions de la classe **Codeur** (p. ??) **Codeur** (p. ??) optqie de 20 pas

Définition à la ligne 9 du fichier Codeur.cpp.

5.4.2.2 **Codeur** : **~Codeur** ( )

Définition à la ligne 15 du fichier Codeur.cpp.

### 5.4.3 Documentation des fonctions membres

#### 5.4.3.1 void Codeur : :compteurRoueCodeuse ( bool ouvFerm )

---compteur roue codeuse---

Définition à la ligne 63 du fichier Codeur.cpp.

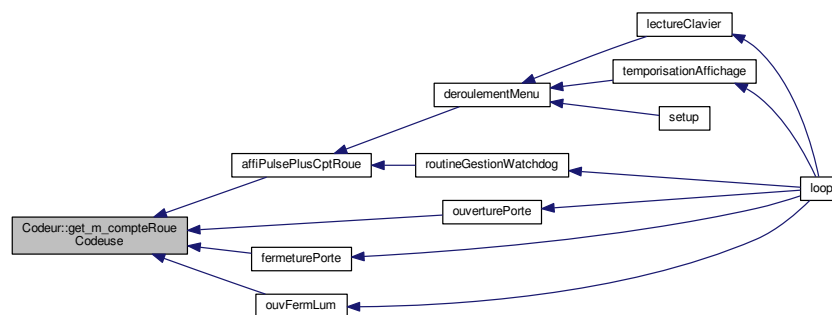
Voici le graphe des appelants de cette fonction :



#### 5.4.3.2 unsigned int Codeur : :get\_m\_compteRoueCodeuse ( )

Définition à la ligne 81 du fichier Codeur.cpp.

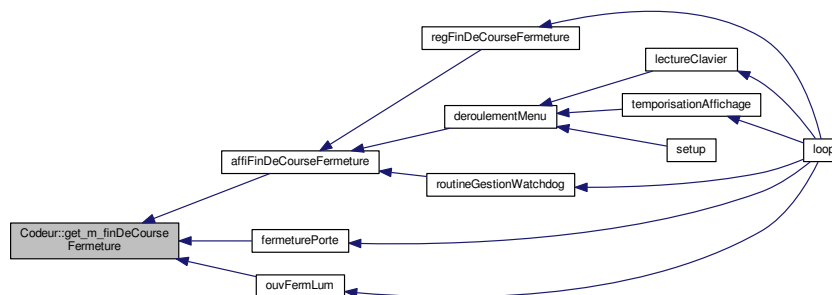
Voici le graphe des appelants de cette fonction :



#### 5.4.3.3 unsigned int Codeur : :get\_m\_finDeCourseFermeture ( )

Définition à la ligne 90 du fichier Codeur.cpp.

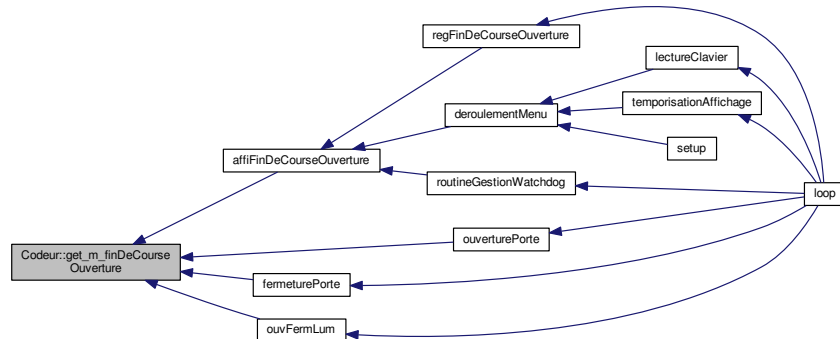
Voici le graphe des appelants de cette fonction :



## 5.4.3.4 unsigned int Codeur : :get\_m\_finDeCourseOuverture ( )

Définition à la ligne 99 du fichier Codeur.cpp.

Voici le graphe des appelants de cette fonction :



## 5.4.3.5 void Codeur : :init ( )

---initialisation---

Définition à la ligne 20 du fichier Codeur.cpp.

Voici le graphe des appelants de cette fonction :

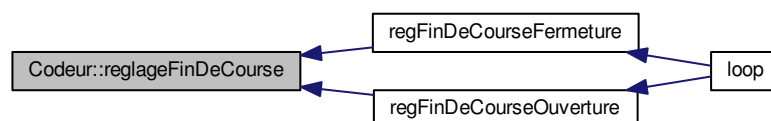


## 5.4.3.6 unsigned int Codeur : :reglageFinDeCourse ( bool ouvFerm, byte touche )

---reglage de la fin de course---

Définition à la ligne 39 du fichier Codeur.cpp.

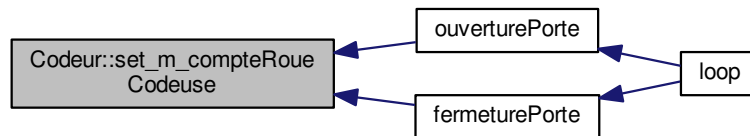
Voici le graphe des appelants de cette fonction :



#### 5.4.3.7 void Codeur : :set\_m\_compteRoueCodeuse ( unsigned int *compteRoueCodeuse* )

Définition à la ligne 85 du fichier Codeur.cpp.

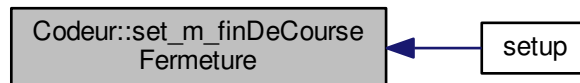
Voici le graphe des appelants de cette fonction :



#### 5.4.3.8 void Codeur : :set\_m\_finDeCourseFermeture ( unsigned int *finDeCourseFermeture* )

Définition à la ligne 94 du fichier Codeur.cpp.

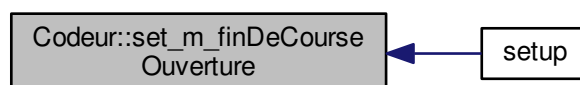
Voici le graphe des appelants de cette fonction :



#### 5.4.3.9 void Codeur : :set\_m\_finDeCourseOuverture ( unsigned int *finDeCourseOuverture* )

Définition à la ligne 103 du fichier Codeur.cpp.

Voici le graphe des appelants de cette fonction :



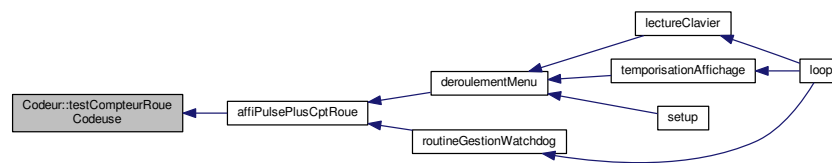


## 5.4.3.10 byte Codeur : testCompteurRoueCodeuse ( byte tolerance )

---test du compteur roue codeuse pour affichage ouvert ou ferme---

Définition à la ligne 26 du fichier Codeur.cpp.

Voici le graphe des appelants de cette fonction :



## 5.4.4 Documentation des données membres

## 5.4.4.1 volatile unsigned int Codeur : m\_compteRoueCodeuse [protected]

Définition à la ligne 37 du fichier Codeur.h.

## 5.4.4.2 const boolean Codeur : m\_debug [protected]

Définition à la ligne 36 du fichier Codeur.h.

## 5.4.4.3 unsigned int Codeur : m\_finDeCourseFermeture [protected]

Définition à la ligne 40 du fichier Codeur.h.

## 5.4.4.4 const unsigned int Codeur : m\_finDeCourseMax [protected]

Définition à la ligne 35 du fichier Codeur.h.

## 5.4.4.5 unsigned int Codeur : m\_finDeCourseOuverture [protected]

Définition à la ligne 41 du fichier Codeur.h.

## 5.4.4.6 volatile boolean Codeur : m\_interruptRoueCodeuse [protected]

Définition à la ligne 38 du fichier Codeur.h.

#### 5.4.4.7 volatile boolean Codeur : :m\_positionRoueCodeuse [protected]

Définition à la ligne 39 du fichier Codeur.h.

#### 5.4.4.8 const byte Codeur : :m\_roueCodeusePin [protected]

Définition à la ligne 34 du fichier Codeur.h.

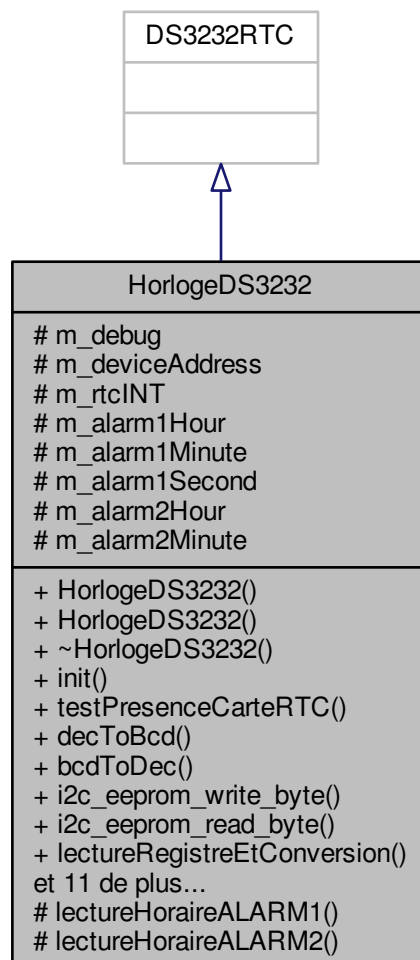
La documentation de cette classe a été générée à partir des fichiers suivants :

```
— /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/Codeur.↔
  h
— /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/Codeur.↔
  cpp
```

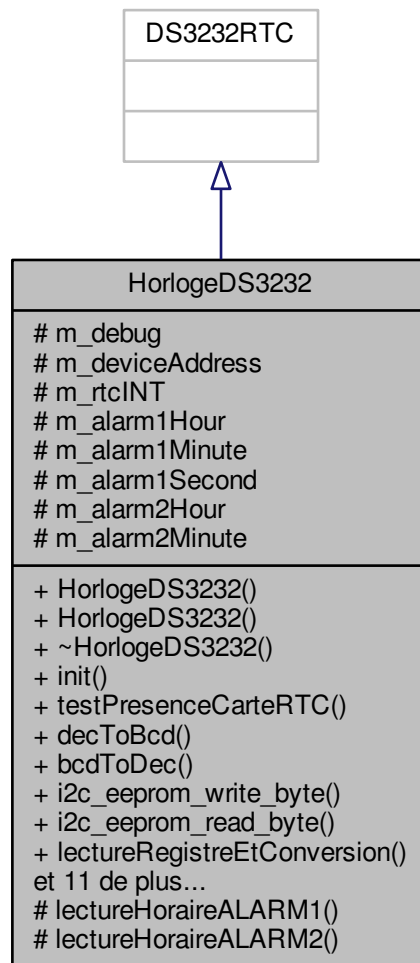
## 5.5 Référence de la classe HorlogeDS3232

```
#include <HorlogeDS3232.h>
```

Graphe d'héritage de HorlogeDS3232 :



Graphe de collaboration de HorlogeDS3232 :



### Fonctions membres publiques

- **HorlogeDS3232** ()
- **HorlogeDS3232** (const byte adresseMemoireI2C=0x57, const byte m\_rtcINT=5, const boolean debug=false)
- **~HorlogeDS3232** ()
- void **init** ()
  - initialisation----
- bool **testPresenceCartertc** ()
  - verification de la presence de la carte RTC / memoire 24C32----
- byte **decToBcd** (byte val)
  - routine decToBcd : Convert normal decimal numbers to binary coded decimal----
- byte **bcdToDec** (byte val)
  - routine bcdToDec : Convert binary coded decimal to normal decimal numbers----
- void **i2c\_eeprom\_write\_byte** (unsigned int eeaddress, byte data)
  - écriture dans l'eeprom at24c32 de la carte rtc----
- byte **i2c\_eeprom\_read\_byte** (unsigned int eeaddress)
  - lecture de l'eeprom at24c32 de la carte rtc----
- byte **lectureRegistreEtConversion** (byte adresse, byte operationAND=0xFF)
  - lecture registre et conversion----

- byte **reglageHeure** (const byte **touche**, byte tmDateTime, const byte type)  
*---reglaged date time---*
- void **testInterruptRTC** (volatile bool &**interruptRTC**)  
*---test IT RTC---*
- void **reglageAlarme** (const byte **touche**, const byte alarme, const byte type)  
*---reglage de l'alarme---*
- void **sauvEepromChoix** (unsigned int valeurChoix, const bool matinSoirOuvFerm, const bool lumiereFin↵  
DeCourse)  
*---sauvegarde dans l'eeeprom I2C le choix de la lumiere ou de la valeur de fin de course---*
- float **calculTemperature** (const bool **typeTemperature**)  
*-----valeur de la temperature en fonction du type---*
- bool **choixTypeOuvertureFermeture** (bool choixOuvertureFermeture, const byte alarme)  
*---mise à jour du choix du type d'ouverture / fermeture---*
- byte **get\_m\_alarm1Hour** ()
- byte **get\_m\_alarm1Minute** ()
- byte **get\_m\_alarm1Second** ()
- byte **get\_m\_alarm2Hour** ()
- byte **get\_m\_alarm2Minute** ()

### Fonctions membres protégées

- void **lectureHoraireALARM1** ()  
*---lecture horaire ALARM1---*
- void **lectureHoraireALARM2** ()  
*---lecture horaire ALARM2---*

### Attributs protégés

- const boolean **m\_debug**
- const int **m\_deviceAddress**
- const byte **m\_rtcINT**
- byte **m\_alarm1Hour**
- byte **m\_alarm1Minute**
- byte **m\_alarm1Second**
- byte **m\_alarm2Hour**
- byte **m\_alarm2Minute**

## 5.5.1 Description détaillée

Définition à la ligne 15 du fichier HorlogeDS3232.h.

## 5.5.2 Documentation des constructeurs et destructeur

### 5.5.2.1 HorlogeDS3232 : :HorlogeDS3232 ( )

Définition à la ligne 8 du fichier HorlogeDS3232.cpp.

### 5.5.2.2 HorlogeDS3232 : :HorlogeDS3232 ( const byte *adresseMemoireI2C* = 0x57, const byte *m\_rtcINT* = 5, const boolean *debug* = false )

Définition à la ligne 15 du fichier HorlogeDS3232.cpp.

## 5.5.2.3 HorlogeDS3232 : ~HorlogeDS3232 ( )

Définition à la ligne 22 du fichier HorlogeDS3232.cpp.

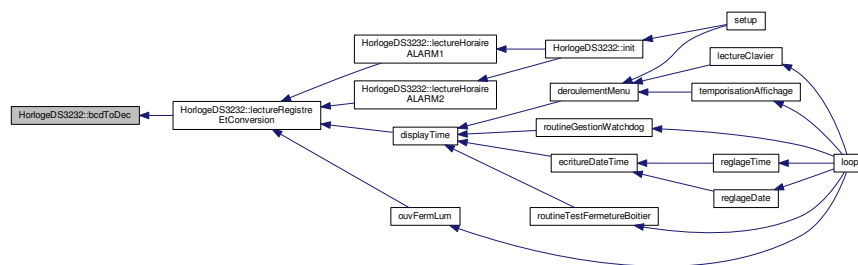
## 5.5.3 Documentation des fonctions membres

## 5.5.3.1 byte HorlogeDS3232 : bcdToDec ( byte val )

-----routine bcdToDec : Convert binary coded decimal to normal decimal numbers-----

Définition à la ligne 57 du fichier HorlogeDS3232.cpp.

Voici le graphe des appelants de cette fonction :

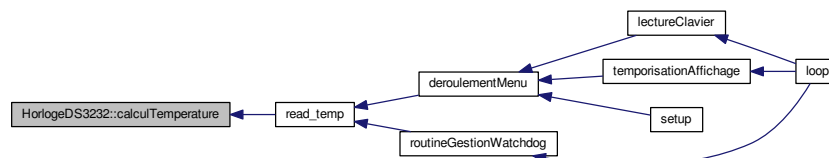


## 5.5.3.2 float HorlogeDS3232 : calculTemperature ( const bool typeTemperature )

-----valeur de la temperature en fonction du type-----

Définition à la ligne 183 du fichier HorlogeDS3232.cpp.

Voici le graphe des appelants de cette fonction :

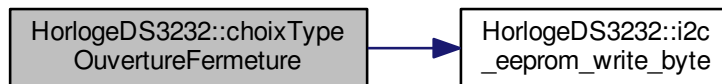


### 5.5.3.3 bool HorlogeDS3232::choixTypeOuvertureFermeture ( bool *choixOuvertureFermeture*, const byte *alarme* )

---mise à jour du choix du type d'ouverture / fermeture---

Définition à la ligne 195 du fichier HorlogeDS3232.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



### 5.5.3.4 byte HorlogeDS3232::decToBcd ( byte *val* )

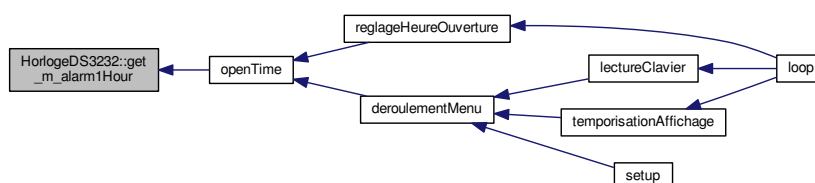
---routine decToBcd : Convert normal decimal numbers to binary coded decimal---

Définition à la ligne 52 du fichier HorlogeDS3232.cpp.

### 5.5.3.5 byte HorlogeDS3232::get\_m\_alarm1Hour ( )

Définition à la ligne 205 du fichier HorlogeDS3232.cpp.

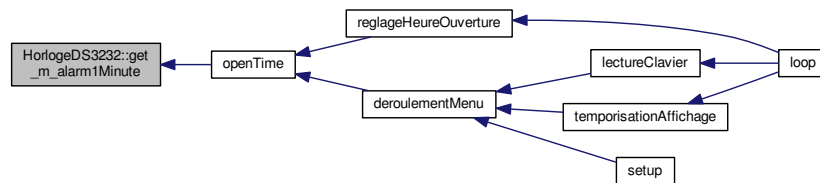
Voici le graphe des appelants de cette fonction :



## 5.5.3.6 byte HorlogeDS3232::get\_m\_alarm1Minute ( )

Définition à la ligne 210 du fichier HorlogeDS3232.cpp.

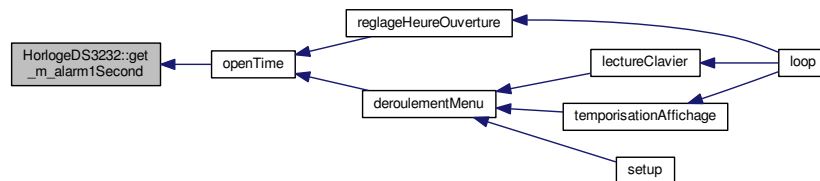
Voici le graphe des appelants de cette fonction :



## 5.5.3.7 byte HorlogeDS3232::get\_m\_alarm1Second ( )

Définition à la ligne 215 du fichier HorlogeDS3232.cpp.

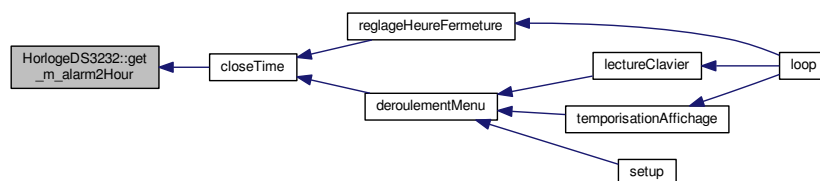
Voici le graphe des appelants de cette fonction :



## 5.5.3.8 byte HorlogeDS3232::get\_m\_alarm2Hour ( )

Définition à la ligne 220 du fichier HorlogeDS3232.cpp.

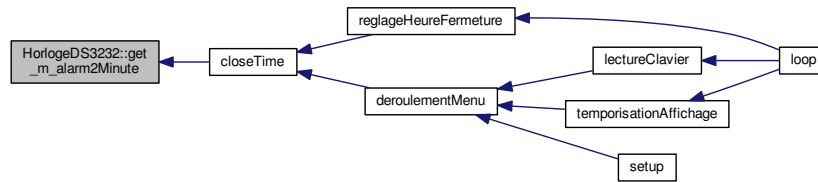
Voici le graphe des appelants de cette fonction :



### 5.5.3.9 byte HorlogeDS3232 : :get\_m\_alarm2Minute ( )

Définition à la ligne 225 du fichier HorlogeDS3232.cpp.

Voici le graphe des appelants de cette fonction :

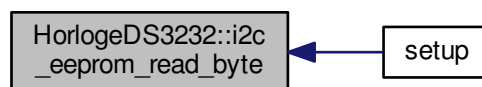


### 5.5.3.10 byte HorlogeDS3232 : :i2c\_eeprom\_read\_byte ( unsigned int eeaddress )

---lecture de l'eeprom at24c32 de la carte rtc---

Définition à la ligne 73 du fichier HorlogeDS3232.cpp.

Voici le graphe des appelants de cette fonction :

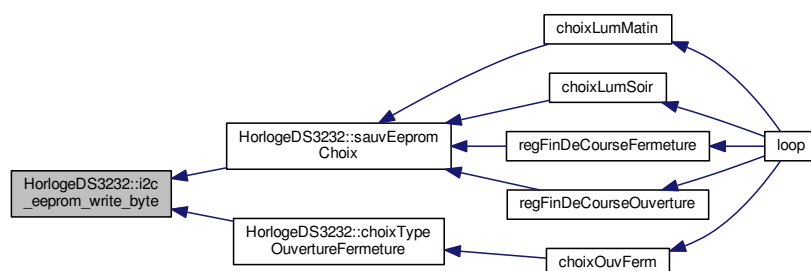


### 5.5.3.11 void HorlogeDS3232 : :i2c\_eeprom\_write\_byte ( unsigned int eeaddress, byte data )

---écriture dans l'eeprom at24c32 de la carte rtc---

Définition à la ligne 63 du fichier HorlogeDS3232.cpp.

Voici le graphe des appelants de cette fonction :



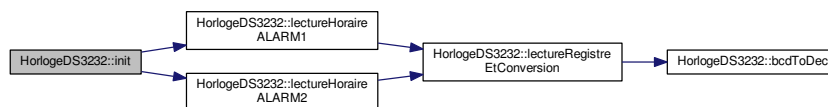


## 5.5.3.12 void HorlogeDS3232 : :init ( )

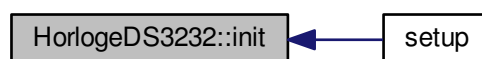
---initialisation---

Définition à la ligne 27 du fichier HorlogeDS3232.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

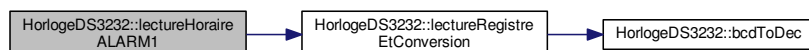


## 5.5.3.13 void HorlogeDS3232 : :lectureHoraireALARM1 ( ) [protected]

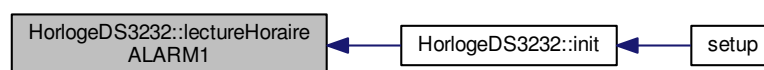
---lecture horaire ALARM1---

Définition à la ligne 125 du fichier HorlogeDS3232.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

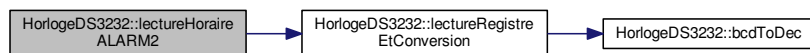


#### 5.5.3.14 void HorlogeDS3232 : :lectureHoraireALARM2 ( ) [protected]

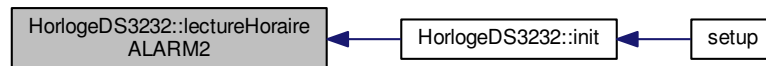
----lecture horaire ALARM2----

Définition à la ligne 132 du fichier HorlogeDS3232.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 5.5.3.15 byte HorlogeDS3232 : :lectureRegistreEtConversion ( byte *adresse*, byte *operation* AND = 0xFF )

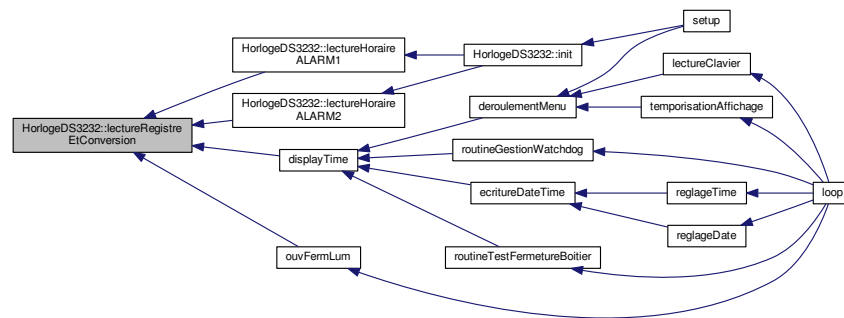
----lecture registre et conversion----

Définition à la ligne 85 du fichier HorlogeDS3232.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 5.5.3.16 void HorlogeDS3232 ::reglageAlarme ( const byte *touche*, const byte *alarme*, const byte *type* )

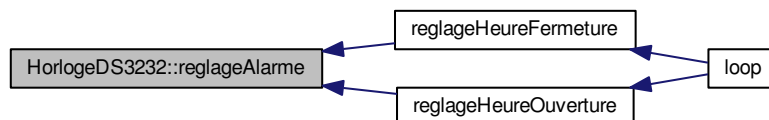
----reglage de l'alarme----

Définition à la ligne 138 du fichier HorlogeDS3232.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

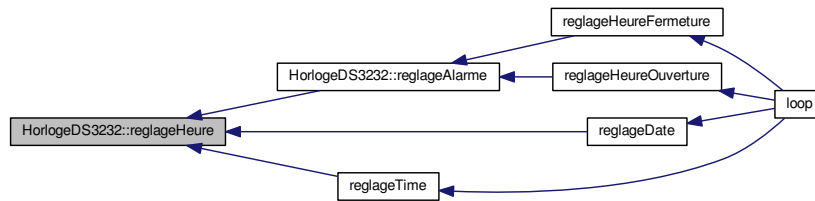


#### 5.5.3.17 byte HorlogeDS3232 ::reglageHeure ( const byte *touche*, byte *tmDateTime*, const byte *type* )

----reglaged date time----

Définition à la ligne 90 du fichier HorlogeDS3232.cpp.

Voici le graphe des appelants de cette fonction :



5.5.3.18 void HorlogeDS3232 : :sauvEepromChoix ( unsigned int *valeurChoix*, const bool *matinSoirOuvFerm*, const bool *lumiereFinDeCourse* )

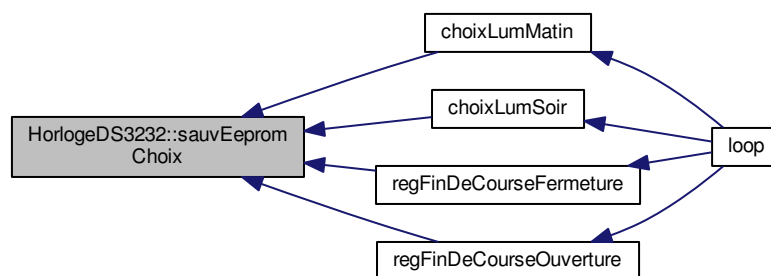
---sauvegarde dans l'eeprom I2C le choix de la lumiere ou de la valeur de fin de course---

Définition à la ligne 167 du fichier HorlogeDS3232.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



5.5.3.19 void HorlogeDS3232 : :testInterruptRTC ( volatile bool & *interruptRTC* )

----test IT RTC----

Définition à la ligne 45 du fichier HorlogeDS3232.cpp.

Voici le graphe des appelants de cette fonction :

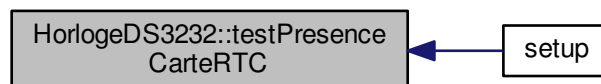


5.5.3.20 bool HorlogeDS3232 : :testPresenceCarteRTC ( )

----verification de la presence de la carte RTC / memoire 24C32----

Définition à la ligne 35 du fichier HorlogeDS3232.cpp.

Voici le graphe des appelants de cette fonction :



## 5.5.4 Documentation des données membres

5.5.4.1 byte HorlogeDS3232 : :m\_alarm1Hour [protected]

Définition à la ligne 55 du fichier HorlogeDS3232.h.

5.5.4.2 byte HorlogeDS3232 : :m\_alarm1Minute [protected]

Définition à la ligne 56 du fichier HorlogeDS3232.h.

5.5.4.3 byte HorlogeDS3232 : :m\_alarm1Second [protected]

Définition à la ligne 57 du fichier HorlogeDS3232.h.

5.5.4.4 `byte HorlogeDS3232 : :m_alarm2Hour` [protected]

Définition à la ligne 58 du fichier HorlogeDS3232.h.

5.5.4.5 `byte HorlogeDS3232 : :m_alarm2Minute` [protected]

Définition à la ligne 59 du fichier HorlogeDS3232.h.

5.5.4.6 `const boolean HorlogeDS3232 : :m_debug` [protected]

Définition à la ligne 52 du fichier HorlogeDS3232.h.

5.5.4.7 `const int HorlogeDS3232 : :m_deviceAddress` [protected]

Définition à la ligne 53 du fichier HorlogeDS3232.h.

5.5.4.8 `const byte HorlogeDS3232 : :m_rtcINT` [protected]

Définition à la ligne 54 du fichier HorlogeDS3232.h.

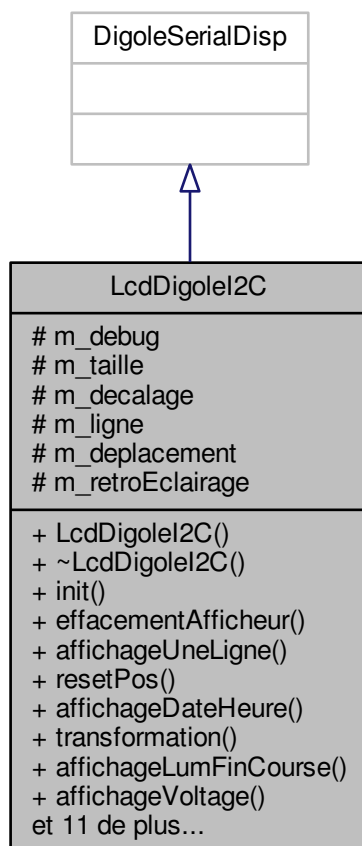
La documentation de cette classe a été générée à partir des fichiers suivants :

- `/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/HorlogeDS3232.h`↔
- `/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/HorlogeDS3232.cpp`↔

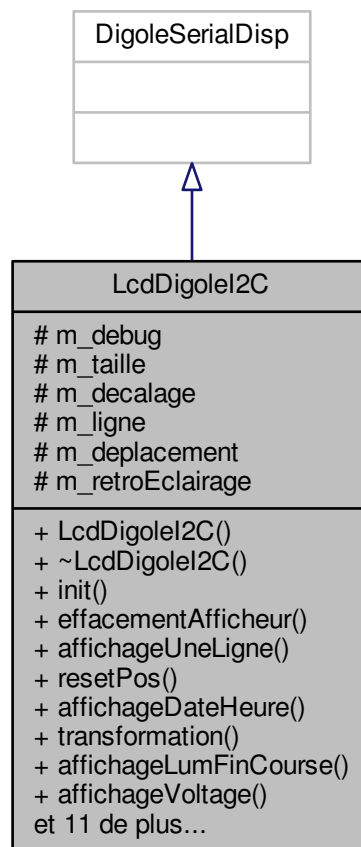
## 5.6 Référence de la classe LcdDigoleI2C

```
#include <LcdDigoleI2C.h>
```

Graphe d'héritage de LcdDigoleI2C :



Graphe de collaboration de LcdDigoleI2C :



## Fonctions membres publiques

- **LcdDigoleI2C** (TwoWire \*, char, byte taille, boolean **debug**=false)
- **~LcdDigoleI2C** ()
- void **init** ()  
    *---initialisation for text LCD adapter---*
- void **effacementAfficheur** ()  
    *---effacement de l'afficheur---*
- void **affichageUneLigne** (String chaine)  
    *---affichage une ligne---*
- void **resetPos** (byte ligne)  
    *---reset display position and clean the line---*
- void **affichageDateHeure** (String **jourSemaine**, byte jourHeure, byte moisMinute, byte anneeSeconde)  
    *---affichage de la date ou de l'heure---*
- String **transformation** (String chaine, byte dateHeure)  
    *---transformation donnees date et heure---*
- void **affichageLumFinCourse** (unsigned int **lum**, byte ligne, bool siNonReglable=0)  
    *---affichage lumiere et fin de course---*
- void **affichageVoltage** (float voltage, String texte, byte ligne)  
    *---affichage tensions---*
- void **affichageChoix** (bool ouverture, bool fermeture, byte ligne)  
    *---affichage choix ouverture fermeture---*
- void **affichageServo** (int pulse, int **compteRoueCodeuse**, byte ligne)



```

    ----affichage pulse et roue codeuse du servo-----
— void razLcd ()
    ----remise à zero du lcd-----
— void bonjour (String chaine1, String chaine2)
    ----affichage au demarrage sur les deux lignes-----
— void cursorPosition (byte decalage, byte ligne, char *texte)
    ----position du curseur : decalage, ligne, texte-----
— void gestionCurseur (bool curseur)
    ----activation / desactivation du curseur-----
— void retroEclairage ()
    ----activation / desactivation du retro eclaire-----
— void choixRetroEclairage (bool choix)
    ----choix activation / desactivation du retro eclaire-----
— void cursorPositionReglages (const byte &toucher, bool &relache, bool &reglage, const byte decalageSup,
    const byte decalageInf)
    ----position du curseur pendant les reglages-----
— byte get_m_decalage ()
— void set_m_decalage (byte decalage)

```

### Attributs protégés

```

— const boolean m_debug
— int const m_taille
— byte m_decalage
— byte m_ligne
— byte m_deplacement
— bool m_retroEclairage

```

#### 5.6.1 Description détaillée

Définition à la ligne 25 du fichier LcdDigoleI2C.h.

#### 5.6.2 Documentation des constructeurs et destructeur

5.6.2.1 **LcdDigoleI2C::LcdDigoleI2C ( TwoWire \*, char , byte *taille*, boolean *debug* = false )**

Définition à la ligne 11 du fichier LcdDigoleI2C.cpp.

5.6.2.2 **LcdDigoleI2C::~~LcdDigoleI2C ( )**

Définition à la ligne 16 du fichier LcdDigoleI2C.cpp.

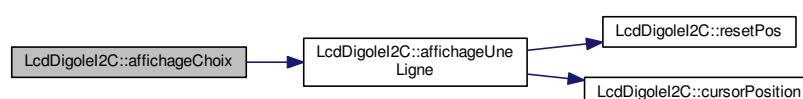
#### 5.6.3 Documentation des fonctions membres

5.6.3.1 **void LcdDigoleI2C::affichageChoix ( bool *ouverture*, bool *fermeture*, byte *ligne* )**

----affichage choix ouverture fermeture----

Définition à la ligne 154 du fichier LcdDigoleI2C.cpp.

Voici le graphe d'appel pour cette fonction :

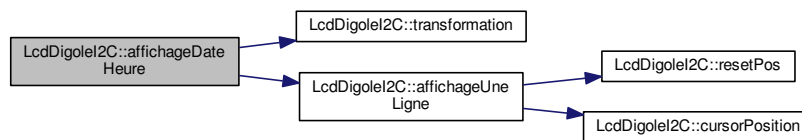


### 5.6.3.2 void LcdDigolel2C : :affichageDateHeure ( String *jourSemaine*, byte *jourHeure*, byte *moisMinute*, byte *anneeSeconde* )

---affichage de la date ou de l'heure---

Définition à la ligne 91 du fichier LcdDigolel2C.cpp.

Voici le graphe d'appel pour cette fonction :

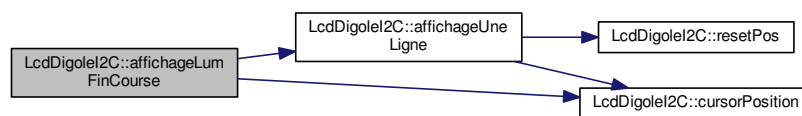


### 5.6.3.3 void LcdDigolel2C : :affichageLumFinCourse ( unsigned int *lum*, byte *ligne*, bool *siNonReglable* = 0 )

---affichage lumiere et fin de course---

Définition à la ligne 131 du fichier LcdDigolel2C.cpp.

Voici le graphe d'appel pour cette fonction :

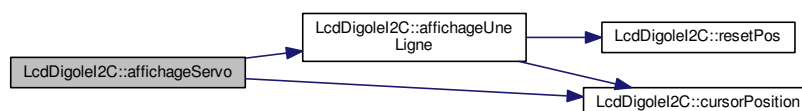


### 5.6.3.4 void LcdDigolel2C : :affichageServo ( int *pulse*, int *compteRoueCodeuse*, byte *ligne* )

---affichage pulse et roue codeuse du servo---

Définition à la ligne 166 du fichier LcdDigolel2C.cpp.

Voici le graphe d'appel pour cette fonction :

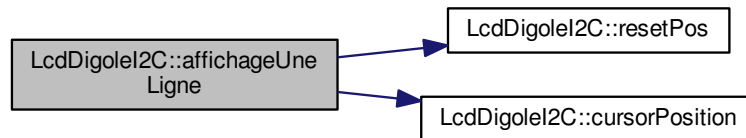


### 5.6.3.5 void LcdDigoleI2C : :affichageUneLigne ( String chaine )

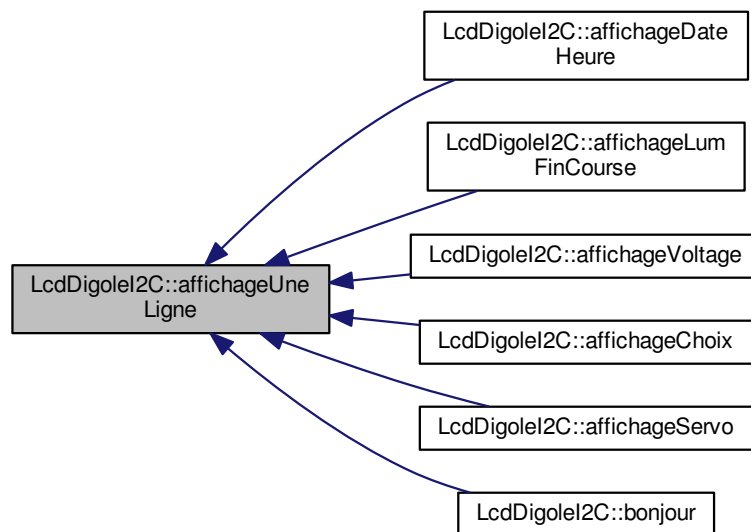
—affichage une ligne—

Définition à la ligne 70 du fichier LcdDigoleI2C.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

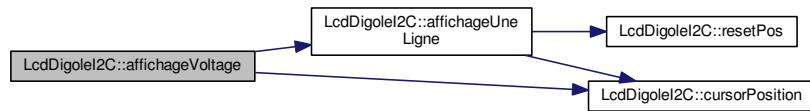


### 5.6.3.6 void LcdDigoleI2C : :affichageVoltage ( float voltage, String texte, byte ligne )

—affichage tensions—

Définition à la ligne 142 du fichier LcdDigoleI2C.cpp.

Voici le graphe d'appel pour cette fonction :

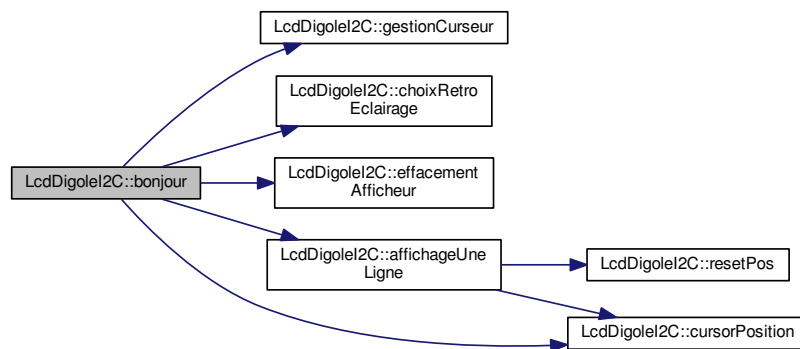


#### 5.6.3.7 void LcdDigolel2C : :bonjour ( String chaine1, String chaine2 )

---affichage au demarrage sur les deux lignes---

Définition à la ligne 179 du fichier LcdDigolel2C.cpp.

Voici le graphe d'appel pour cette fonction :

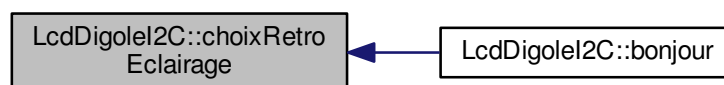


#### 5.6.3.8 void LcdDigolel2C : :choixRetroEclairage ( bool choix )

---choix activation / desactivation du retro eclaireage---

Définition à la ligne 59 du fichier LcdDigolel2C.cpp.

Voici le graphe des appelants de cette fonction :

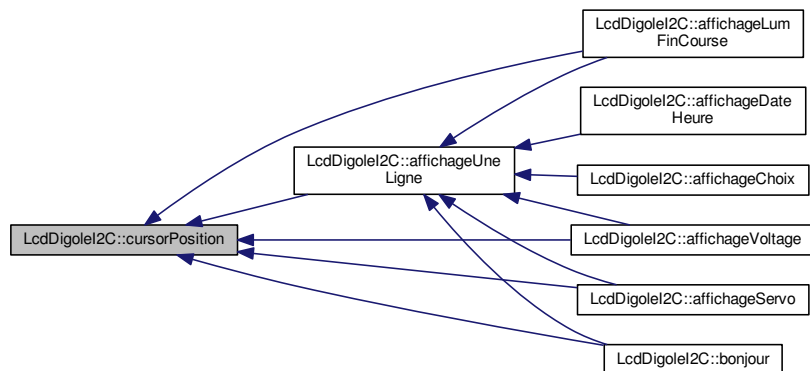


### 5.6.3.9 void LcdDigoleI2C::cursorPosition ( byte *decalage*, byte *ligne*, char \* *texte* )

——position du curseur : *decalage*, *ligne*, *texte*——

Définition à la ligne 202 du fichier LcdDigoleI2C.cpp.

Voici le graphe des appelants de cette fonction :



### 5.6.3.10 void LcdDigoleI2C::cursorPositionReglages ( const byte & *touche*, bool & *relache*, bool & *reglage*, const byte *decalageSup*, const byte *deplacement*, const byte *decalageInf* )

——position du curseur pendant les réglages——

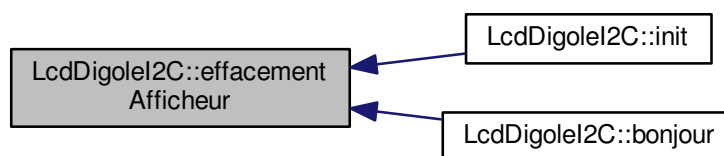
Définition à la ligne 208 du fichier LcdDigoleI2C.cpp.

### 5.6.3.11 void LcdDigoleI2C::effacementAfficheur ( )

——effacement de l'afficheur——

Définition à la ligne 36 du fichier LcdDigoleI2C.cpp.

Voici le graphe des appelants de cette fonction :



#### 5.6.3.12 void LcdDigoIeI2C : :gestionCurseur ( bool *curseur* )

-----activation / desactivation du curseur-----

Définition à la ligne 197 du fichier LcdDigoIeI2C.cpp.

Voici le graphe des appelants de cette fonction :



#### 5.6.3.13 byte LcdDigoIeI2C : :get\_m\_decalage ( )

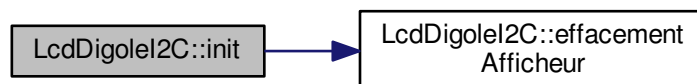
Définition à la ligne 223 du fichier LcdDigoIeI2C.cpp.

#### 5.6.3.14 void LcdDigoIeI2C : :init ( )

-----initialisation for text LCD adapter-----

Définition à la ligne 21 du fichier LcdDigoIeI2C.cpp.

Voici le graphe d'appel pour cette fonction :



#### 5.6.3.15 void LcdDigoIeI2C : :razLcd ( )

-----remise à zero du lcd-----

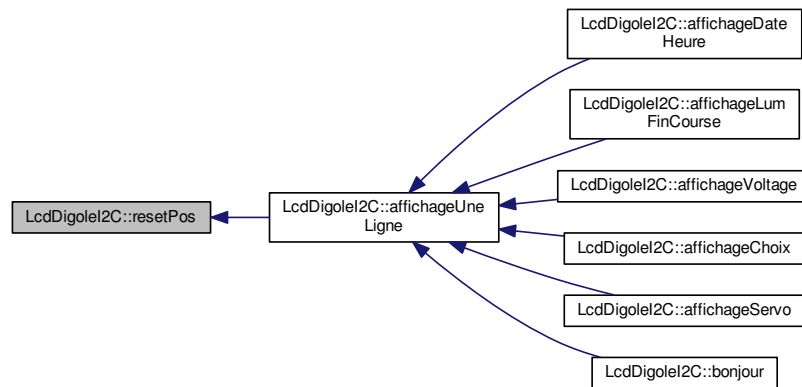
Définition à la ligne 41 du fichier LcdDigoIeI2C.cpp.

#### 5.6.3.16 void LcdDigoleI2C : :resetPos ( byte *ligne* )

---reset display position and clean the line---

Définition à la ligne 79 du fichier LcdDigoleI2C.cpp.

Voici le graphe des appelants de cette fonction :



#### 5.6.3.17 void LcdDigoleI2C : :retroEclairage ( )

---activation / desactivation du retro eclairage---

Définition à la ligne 48 du fichier LcdDigoleI2C.cpp.

#### 5.6.3.18 void LcdDigoleI2C : :set\_m\_decalage ( byte *decalage* )

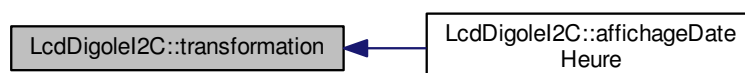
Définition à la ligne 228 du fichier LcdDigoleI2C.cpp.

#### 5.6.3.19 String LcdDigoleI2C : :transformation ( String *chaine*, byte *dateHeure* )

---transformation donnees date et heure---

Définition à la ligne 113 du fichier LcdDigoleI2C.cpp.

Voici le graphe des appelants de cette fonction :



### 5.6.4 Documentation des données membres

5.6.4.1 `const boolean LcdDigolel2C : :m_debug` [protected]

Définition à la ligne 57 du fichier LcdDigolel2C.h.

5.6.4.2 `byte LcdDigolel2C : :m_decalage` [protected]

Définition à la ligne 59 du fichier LcdDigolel2C.h.

5.6.4.3 `byte LcdDigolel2C : :m_deplacement` [protected]

Définition à la ligne 61 du fichier LcdDigolel2C.h.

5.6.4.4 `byte LcdDigolel2C : :m_ligne` [protected]

Définition à la ligne 60 du fichier LcdDigolel2C.h.

5.6.4.5 `bool LcdDigolel2C : :m_retroEclairage` [protected]

Définition à la ligne 62 du fichier LcdDigolel2C.h.

5.6.4.6 `int const LcdDigolel2C : :m_taille` [protected]

Définition à la ligne 58 du fichier LcdDigolel2C.h.

La documentation de cette classe a été générée à partir des fichiers suivants :

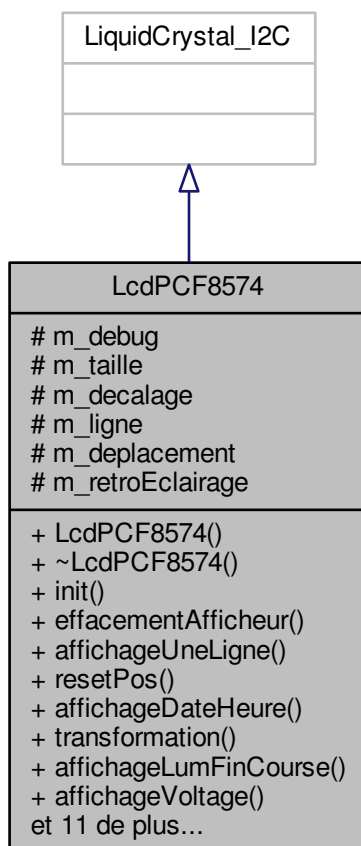
- /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/**LcdDigolel2C.h**
- /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/**LcdDigolel2C.cpp**



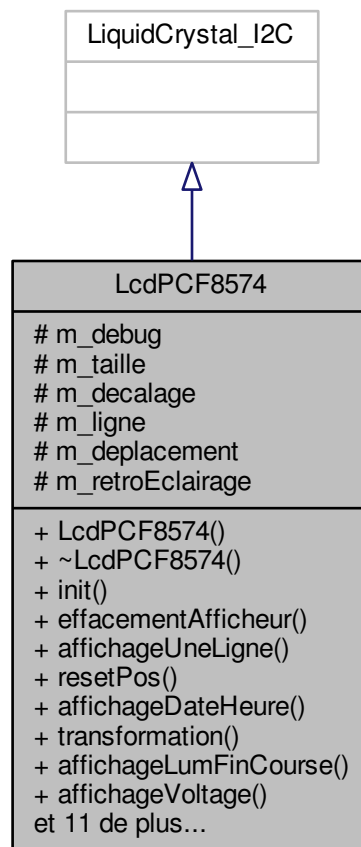
## 5.7 Référence de la classe LcdPCF8574

```
#include <LcdPCF8574.h>
```

Graphe d'héritage de LcdPCF8574 :



Graphe de collaboration de LcdPCF8574 :



## Fonctions membres publiques

- **LcdPCF8574** (char, const byte taille=16, const byte nbLigne=2, boolean **debug**=false)
- **~LcdPCF8574** ()
- void **init** ()
  - initialisation for text LCD adapter—
- void **effacementAfficheur** ()
  - effacement de l'afficheur—
- void **affichageUneLigne** (String chaine)
  - affichage une ligne—
- void **resetPos** (byte ligne)
  - reset display position and clean the line—
- void **affichageDateHeure** (String **jourSemaine**, byte jourHeure, byte moisMinute, byte anneeSeconde)
  - affichage de la date ou de l'heure—
- String **transformation** (String chaine, byte dateHeure)
  - transformation donnees date et heure—
- void **affichageLumFinCourse** (unsigned int **lum**, byte ligne, bool siNonReglable=0)
  - affichage lumiere et fin de course—
- void **affichageVoltage** (float voltage, String texte, byte ligne)
  - affichage tensions—
- void **affichageChoix** (bool ouverture, bool fermeture, byte ligne)
  - affichage choix ouverture fermeture—
- void **affichageServo** (int pulse, int **compteRoueCodeuse**, byte ligne)

```

    ----affichage pulse et roue codeuse du servo-----
— void razLcd ()
    ----remise à zero du lcd-----
— void bonjour (String chaine1, String chaine2)
    ----Bonjour-----
— void cursorPosition (byte decalage, byte ligne, char *texte)
    ----position du curseur : decalage, ligne, texte-----
— void gestionCurseur (bool curseur)
    ----activation / desactivation du curseur-----
— void retroEclairage ()
    ----activation / desactivation du retro eclaire-----
— void choixRetroEclairage (bool choix)
    ----choix activation / desactivation du retro eclaire-----
— void cursorPositionReglages (const byte &touche, bool &relache, bool &reglage, const byte decalageSup,
    const byte deplacement, const byte decalageInf)
    ----position du cuseur pendant les reglages-----
— byte get_m_decalage ()
— void set_m_decalage (byte decalage)

```

### Attributs protégés

```

— const boolean m_debug
— int const m_taille
— byte m_decalage
— byte m_ligne
— byte m_deplacement
— bool m_retroEclairage

```

#### 5.7.1 Description détaillée

**LcdPCF8574.h** (p. ??) prototype de la classe LCD circuit PCF8574 I2C afficheur lcd 2\*16 caractères avec liaison serie i2c LiquidCrystal\_I2C

Définition à la ligne 13 du fichier LcdPCF8574.h.

#### 5.7.2 Documentation des constructeurs et destructeur

5.7.2.1 **LcdPCF8574** : **LcdPCF8574** ( char , const byte *taille* = 16, const byte *nbLigne* = 2, boolean *debug* = false )

Définition à la ligne 13 du fichier LcdPCF8574.cpp.

5.7.2.2 **LcdPCF8574** : **~LcdPCF8574** ( )

Définition à la ligne 19 du fichier LcdPCF8574.cpp.

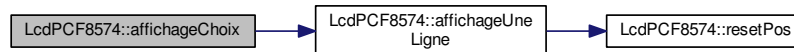
### 5.7.3 Documentation des fonctions membres

#### 5.7.3.1 void LcdPCF8574 : :affichageChoix ( bool ouverture, bool fermeture, byte ligne )

---affichage choix ouverture fermeture---

Définition à la ligne 156 du fichier LcdPCF8574.cpp.

Voici le graphe d'appel pour cette fonction :

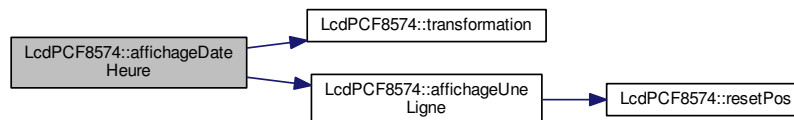


#### 5.7.3.2 void LcdPCF8574 : :affichageDateHeure ( String jourSemaine, byte jourHeure, byte moisMinute, byte anneeSeconde )

---affichage de la date ou de l'heure---

Définition à la ligne 91 du fichier LcdPCF8574.cpp.

Voici le graphe d'appel pour cette fonction :

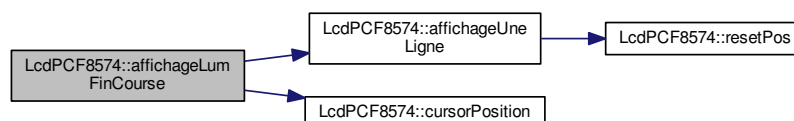


#### 5.7.3.3 void LcdPCF8574 : :affichageLumFinCourse ( unsigned int lum, byte ligne, bool siNonReglable = 0 )

---affichage lumiere et fin de course---

Définition à la ligne 133 du fichier LcdPCF8574.cpp.

Voici le graphe d'appel pour cette fonction :

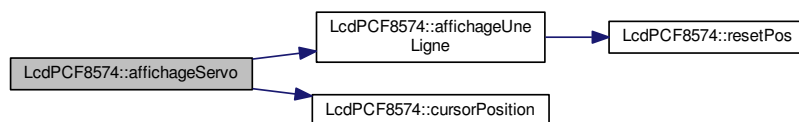


#### 5.7.3.4 void LcdPCF8574 : :affichageServo ( int *pulse*, int *compteRoueCodeuse*, byte *ligne* )

----affichage pulse et roue codeuse du servo-----

Définition à la ligne 168 du fichier LcdPCF8574.cpp.

Voici le graphe d'appel pour cette fonction :



#### 5.7.3.5 void LcdPCF8574 : :affichageUneLigne ( String *chaine* )

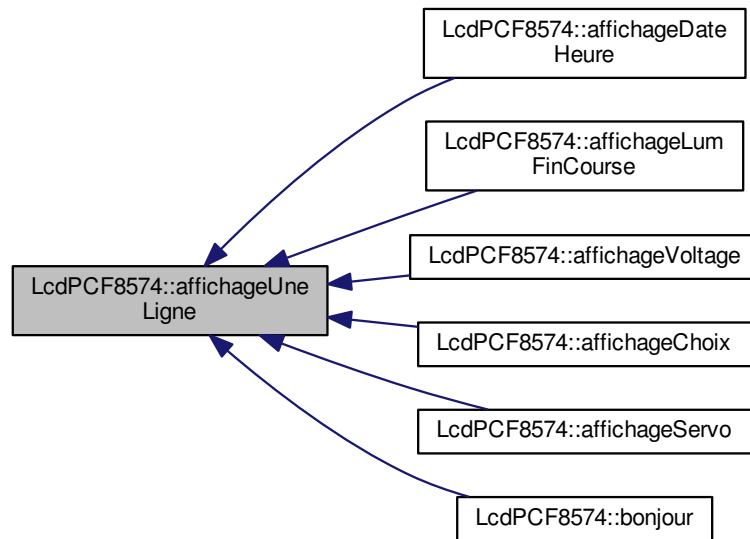
----affichage une ligne----

Définition à la ligne 69 du fichier LcdPCF8574.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

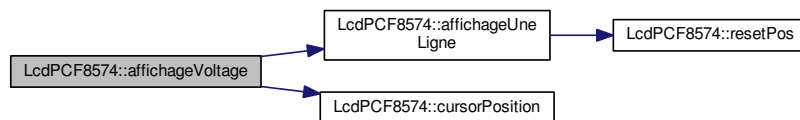


#### 5.7.3.6 void LcdPCF8574 : :affichageVoltage ( float voltage, String texte, byte ligne )

---affichage tensions---

Définition à la ligne 144 du fichier `LcdPCF8574.cpp`.

Voici le graphe d'appel pour cette fonction :

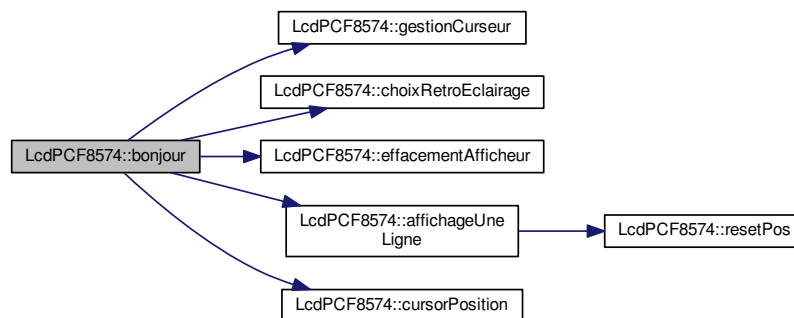


#### 5.7.3.7 void LcdPCF8574 : :bonjour ( String chaine1, String chaine2 )

---Bonjour---

Définition à la ligne 181 du fichier `LcdPCF8574.cpp`.

Voici le graphe d'appel pour cette fonction :

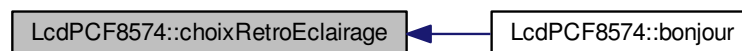


#### 5.7.3.8 void LcdPCF8574 : :choixRetroEclairage ( bool choix )

----choix activation / desactivation du retro eclaireage----

Définition à la ligne 58 du fichier LcdPCF8574.cpp.

Voici le graphe des appelants de cette fonction :

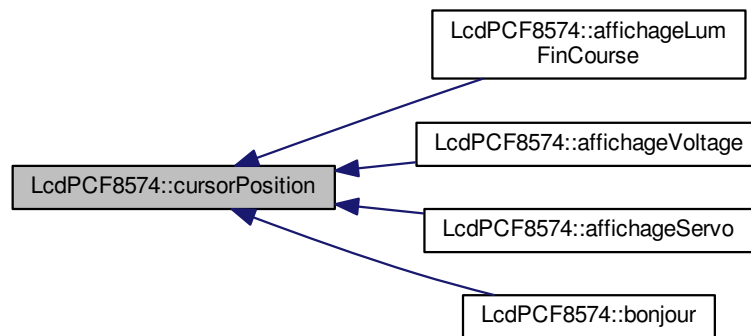


#### 5.7.3.9 void LcdPCF8574 : :cursorPosition ( byte decalage, byte ligne, char \* texte )

----position du curseur : decalage, ligne, texte----

Définition à la ligne 204 du fichier LcdPCF8574.cpp.

Voici le graphe des appelants de cette fonction :



5.7.3.10 `void LcdPCF8574 : :cursorPositionReglages ( const byte & touche, bool & relache, bool & reglage, const byte decalageSup, const byte deplacement, const byte decalageInf )`

——position du cuseur pendant les reglages——

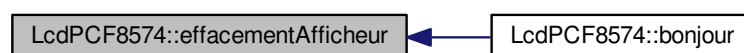
Définition à la ligne 210 du fichier `LcdPCF8574.cpp`.

5.7.3.11 `void LcdPCF8574 : :effacementAfficheur ( )`

——effacement de l'afficheur——

Définition à la ligne 35 du fichier `LcdPCF8574.cpp`.

Voici le graphe des appelants de cette fonction :





5.7.3.12 void LcdPCF8574::gestionCurseur ( bool *curseur* )

----activation / desactivation du curseur----

Définition à la ligne 199 du fichier LcdPCF8574.cpp.

Voici le graphe des appelants de cette fonction :



## 5.7.3.13 byte LcdPCF8574::get\_m\_decalage ( )

Définition à la ligne 225 du fichier LcdPCF8574.cpp.

## 5.7.3.14 void LcdPCF8574::init ( )

----initialisation for text LCD adapter----

Définition à la ligne 24 du fichier LcdPCF8574.cpp.

## 5.7.3.15 void LcdPCF8574::razLcd ( )

----remise à zero du lcd----

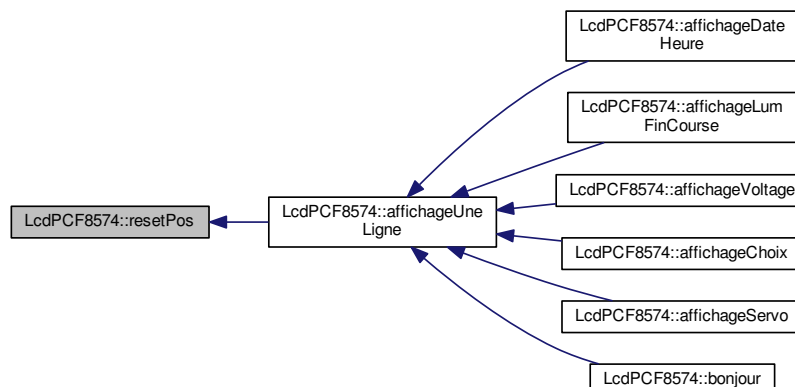
Définition à la ligne 40 du fichier LcdPCF8574.cpp.

5.7.3.16 void LcdPCF8574::resetPos ( byte *ligne* )

----reset display position and clean the line----

Définition à la ligne 79 du fichier LcdPCF8574.cpp.

Voici le graphe des appelants de cette fonction :



#### 5.7.3.17 void LcdPCF8574 : :retroEclairage ( )

----activation / desactivation du retro eclaireage----

Définition à la ligne 47 du fichier LcdPCF8574.cpp.

#### 5.7.3.18 void LcdPCF8574 : :set\_m\_decalage ( byte *decalage* )

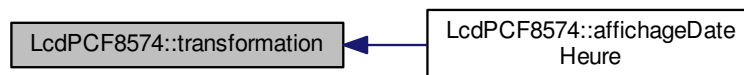
Définition à la ligne 230 du fichier LcdPCF8574.cpp.

#### 5.7.3.19 String LcdPCF8574 : :transformation ( String *chaine*, byte *dateHeure* )

----transformation donnees date et heure----

Définition à la ligne 114 du fichier LcdPCF8574.cpp.

Voici le graphe des appelants de cette fonction :



### 5.7.4 Documentation des données membres

#### 5.7.4.1 const boolean LcdPCF8574 : :m\_debug [protected]

Définition à la ligne 48 du fichier LcdPCF8574.h.

#### 5.7.4.2 byte LcdPCF8574 : :m\_decalage [protected]

Définition à la ligne 50 du fichier LcdPCF8574.h.

#### 5.7.4.3 byte LcdPCF8574 : :m\_deplacement [protected]

Définition à la ligne 52 du fichier LcdPCF8574.h.

#### 5.7.4.4 byte LcdPCF8574 : :m\_ligne [protected]

Définition à la ligne 51 du fichier LcdPCF8574.h.

5.7.4.5 `bool LcdPCF8574 : :m_retroEclairage` [protected]

Définition à la ligne 53 du fichier LcdPCF8574.h.

5.7.4.6 `int const LcdPCF8574 : :m_taille` [protected]

Définition à la ligne 49 du fichier LcdPCF8574.h.

La documentation de cette classe a été générée à partir des fichiers suivants :

- /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/**LcdPCF8574.h**
- /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/**LcdPCF8574.cpp**

## 5.8 Référence de la classe Lumiere

```
#include <Lumiere.h>
```

Graphe de collaboration de Lumiere :

Lumiere
<pre># m_lumMatin # m_lumSoir # m_ouverture # m_fermeture # m_compteurWatchdogLumiere # m_lumiereMax # m_incrementation # m_lumierePin # m_rapportConversion # m_maxCAD # m_heureFenetreSoir # m_tempsLum # m_debug</pre>
<pre>+ Lumiere() + ~Lumiere() + reglageLumiere() + luminositeCAD() + tensionLuminosite() + tensionLuminositeCADvers Float() + testLuminosite() + fenetreNonDeclenchement() + nonDeclenchementPosition Servo() + declenchementServoLuminosite() et 10 de plus...</pre>

## Fonctions membres publiques

- **Lumiere** (const byte **lumierePin**, unsigned int **lumMatin**, unsigned int **lumSoir**, const byte **heureFenetre**↵  
**Soir**=17, const float **rapportConversion**=5, const byte **tempsLum**=2, boolean **debug**=false)
- **~Lumiere** ()
- unsigned int **reglageLumiere** (bool **matinSoir**, byte **touche**)  
    *----reglage de la lumiere du matin ou du soir----*
- int **luminositeCAD** ()  
    *----- lecture luminosite CAD-----*
- float **tensionLuminosite** (int **valLumiere**)  
    *----- conversion CAD vers tension luminosite -----*
- float **tensionLuminositeCADversFloat** ()  
    *----lecture et conversion vers tension luminosite en float----*
- void **testLuminosite** ()  
    *----test luminosite et mise à jour du compteur watchdog lumiere----*
- void **fenetreNonDeclenchement** (byte **horaire**)  
    *----fenetre de non declenchement et mise à jour du compteur watchdog lumiere----*
- void **nonDeclenchementPositionServo** (volatile unsigned int **compteRoueCodeuse**, unsigned int **finDe**↵  
**CourseFermeture**, unsigned int **finDeCourseOuverture**)  
    *----non Declenchement Position Servo et mise à jour du compteur watchdog lumiere----*
- byte **declenchementServoLuminosite** ()  
    *----declenchement servo par la luminosite----*
- bool **get\_m\_ouverture** ()
- void **set\_m\_ouverture** (bool **ouverture**)
- bool **get\_m\_fermeture** ()  
    *----accesseur - getter----*
- void **set\_m\_fermeture** (bool **fermeture**)
- unsigned int **get\_m\_lumMatin** ()
- void **set\_m\_lumMatin** (unsigned int **lumMatin**)
- unsigned int **get\_m\_lumSoir** ()
- void **set\_m\_lumSoir** (unsigned int **lumSoir**)
- unsigned int **get\_m\_compteurWatchdogLumiere** ()
- void **set\_m\_compteurWatchdogLumiere** (unsigned int **compteurWatchdogLumiere**)

## Attributs protégés

- unsigned int **m\_lumMatin**
- unsigned int **m\_lumSoir**  
    *valeur de la lumière du matin*
- bool **m\_ouverture**
- bool **m\_fermeture**
- unsigned int **m\_compteurWatchdogLumiere**
- const unsigned int **m\_lumiereMax**
- const byte **m\_incrementation**
- const byte **m\_lumierePin**
- const float **m\_rapportConversion**
- const int **m\_maxCAD**
- const byte **m\_heureFenetreSoir**
- const byte **m\_tempsLum**
- const boolean **m\_debug**

### 5.8.1 Description détaillée

**Lumiere.h** (p. ??) prototype de la classe **Lumiere** (p. ??) ldr

Définition à la ligne 11 du fichier **Lumiere.h**.

### 5.8.2 Documentation des constructeurs et destructeur

5.8.2.1 **Lumiere** : :**Lumiere** ( const byte *lumierePin*, unsigned int *lumMatin*, unsigned int *lumSoir*, const byte *heureFenetreSoir*  
= 17, const float *rapportConversion* = 5, const byte *tempsLum* = 2, boolean *debug* = false )

**Lumiere.cpp** (p. ??) définitions de la classe **Lumiere** (p. ??) LDR

Définition à la ligne 9 du fichier **Lumiere.cpp**.

## 5.8.2.2 Lumiere : :~Lumiere ( )

Définition à la ligne 17 du fichier Lumiere.cpp.

## 5.8.3 Documentation des fonctions membres

## 5.8.3.1 byte Lumiere : :declenchementServoLuminosite ( )

---declenchement servo par la luminosite---

Définition à la ligne 88 du fichier Lumiere.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

5.8.3.2 void Lumiere : :fenetreNonDeclenchement ( byte *horaire* )

---fenetre de non declenchement et mise à jour du compteur watchdog lumiere---

Définition à la ligne 74 du fichier Lumiere.cpp.

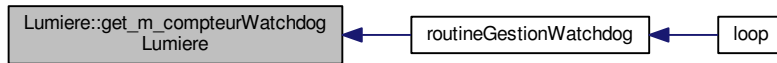
Voici le graphe des appelants de cette fonction :



### 5.8.3.3 unsigned int Lumiere : :get\_m\_compteurWatchdogLumiere ( )

Définition à la ligne 140 du fichier Lumiere.cpp.

Voici le graphe des appelants de cette fonction :

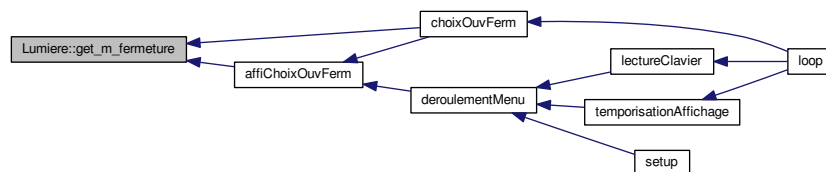


### 5.8.3.4 bool Lumiere : :get\_m\_fermeture ( )

---accesseur - getter---

Définition à la ligne 113 du fichier Lumiere.cpp.

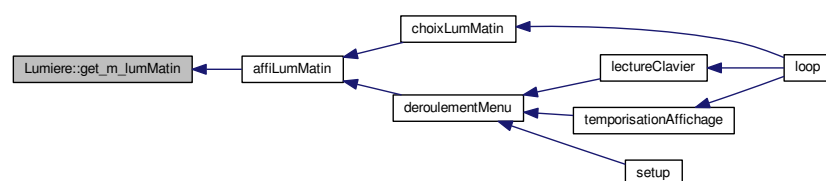
Voici le graphe des appelants de cette fonction :



### 5.8.3.5 unsigned int Lumiere : :get\_m\_lumMatin ( )

Définition à la ligne 122 du fichier Lumiere.cpp.

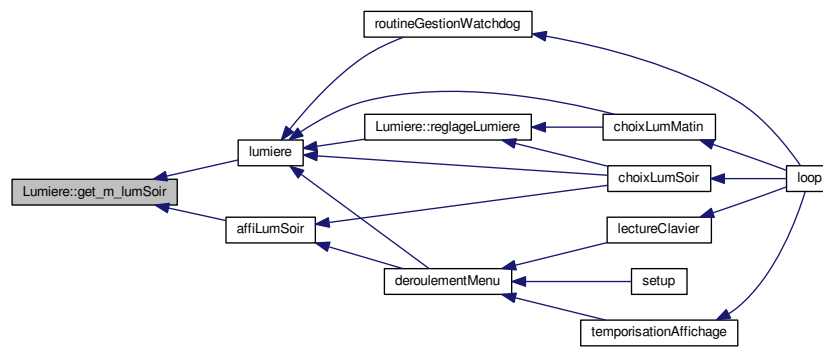
Voici le graphe des appelants de cette fonction :



## 5.8.3.6 unsigned int Lumiere : :get\_m\_lumSoir ( )

Définition à la ligne 131 du fichier Lumiere.cpp.

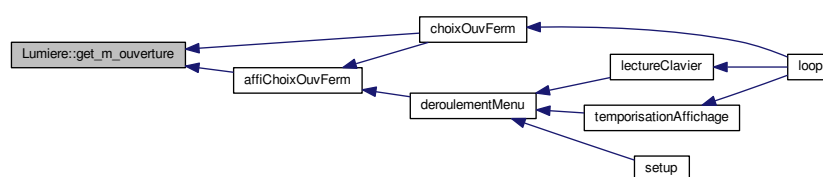
Voici le graphe des appelants de cette fonction :



## 5.8.3.7 bool Lumiere : :get\_m\_ouverture ( )

Définition à la ligne 104 du fichier Lumiere.cpp.

Voici le graphe des appelants de cette fonction :

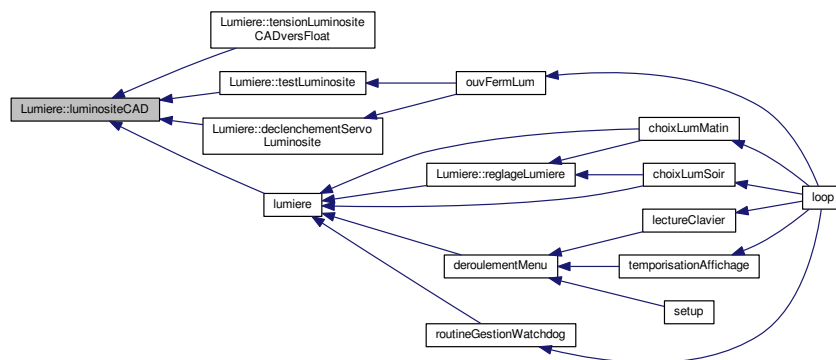


## 5.8.3.8 int Lumiere : :luminositeCAD ( )

----- lecture luminosite CAD-----

Définition à la ligne 46 du fichier Lumiere.cpp.

Voici le graphe des appelants de cette fonction :



#### 5.8.3.9 void Lumiere : :nonDeclenchementPositionServo ( volatile unsigned int *compteRoueCodeuse*, unsigned int *finDeCourseFermeture*, unsigned int *finDeCourseOuverture* )

---non Declenchement Position Servo et mise à jour du compteur watchdog lumiere---

Définition à la ligne 81 du fichier Lumiere.cpp.

Voici le graphe des appelants de cette fonction :

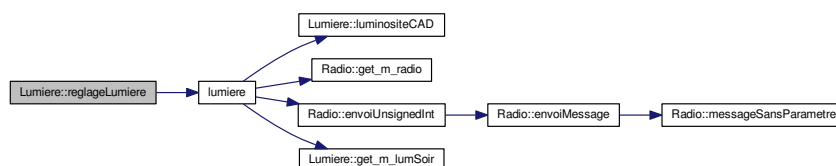


#### 5.8.3.10 unsigned int Lumiere : :reglageLumiere ( bool *matinSoir*, byte *touche* )

---reglage de la lumiere du matin ou du soir---

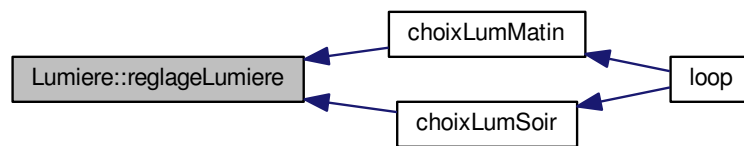
Définition à la ligne 22 du fichier Lumiere.cpp.

Voici le graphe d'appel pour cette fonction :





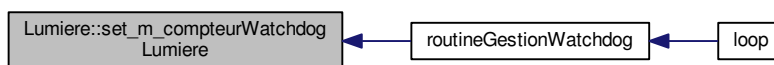
Voici le graphe des appelants de cette fonction :



#### 5.8.3.11 void Lumiere : :set\_m\_compteurWatchdogLumiere ( unsigned int *compteurWatchdogLumiere* )

Définition à la ligne 144 du fichier Lumiere.cpp.

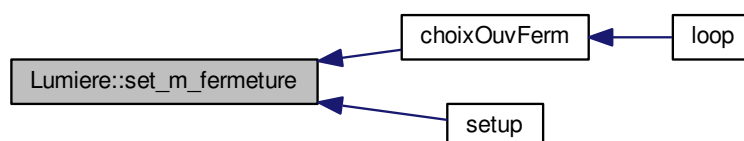
Voici le graphe des appelants de cette fonction :



#### 5.8.3.12 void Lumiere : :set\_m\_fermeture ( bool *fermeture* )

Définition à la ligne 117 du fichier Lumiere.cpp.

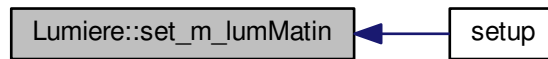
Voici le graphe des appelants de cette fonction :



#### 5.8.3.13 void Lumiere : :set\_m\_lumMatin ( unsigned int *lumMatin* )

Définition à la ligne 126 du fichier Lumiere.cpp.

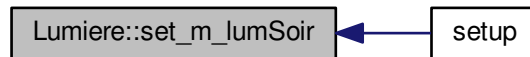
Voici le graphe des appelants de cette fonction :



#### 5.8.3.14 void Lumiere : :set\_m\_lumSoir ( unsigned int *lumSoir* )

Définition à la ligne 135 du fichier Lumiere.cpp.

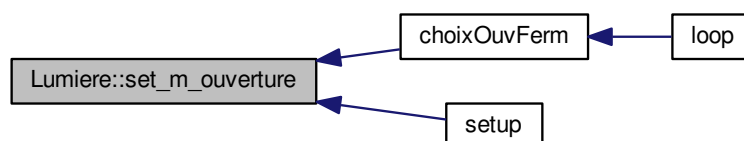
Voici le graphe des appelants de cette fonction :



#### 5.8.3.15 void Lumiere : :set\_m\_ouverture ( bool *ouverture* )

Définition à la ligne 108 du fichier Lumiere.cpp.

Voici le graphe des appelants de cette fonction :

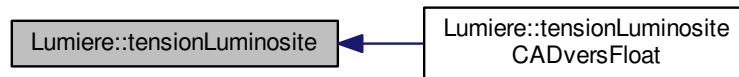


**5.8.3.16** float Lumiere : :tensionLuminosite ( int *valLumiere* )

----- conversion CAD vers tension luminosite -----

Définition à la ligne 52 du fichier Lumiere.cpp.

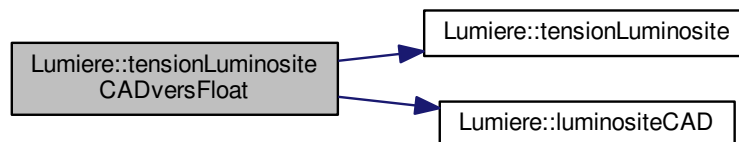
Voici le graphe des appelants de cette fonction :

**5.8.3.17** float Lumiere : :tensionLuminositeCADversFloat ( )

----lecture et conversion vers tension luminosite en float----

Définition à la ligne 59 du fichier Lumiere.cpp.

Voici le graphe d'appel pour cette fonction :

**5.8.3.18** void Lumiere : :testLuminosite ( )

----test luminosite et mise à jour du compteur watchdog lumiere----

Définition à la ligne 65 du fichier Lumiere.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



## 5.8.4 Documentation des données membres

### 5.8.4.1 `unsigned int Lumiere : :m_compteurWatchdogLumiere` [protected]

Définition à la ligne 48 du fichier Lumiere.h.

### 5.8.4.2 `const boolean Lumiere : :m_debug` [protected]

Définition à la ligne 57 du fichier Lumiere.h.

### 5.8.4.3 `bool Lumiere : :m_fermeture` [protected]

Définition à la ligne 47 du fichier Lumiere.h.

### 5.8.4.4 `const byte Lumiere : :m_heureFenetreSoir` [protected]

Définition à la ligne 55 du fichier Lumiere.h.

### 5.8.4.5 `const byte Lumiere : :m_incrementation` [protected]

Définition à la ligne 50 du fichier Lumiere.h.

### 5.8.4.6 `const unsigned int Lumiere : :m_lumiereMax` [protected]

Définition à la ligne 49 du fichier Lumiere.h.

### 5.8.4.7 `const byte Lumiere : :m_lumierePin` [protected]

Définition à la ligne 51 du fichier Lumiere.h.

### 5.8.4.8 `unsigned int Lumiere : :m_lumMatin` [protected]

Définition à la ligne 44 du fichier Lumiere.h.

5.8.4.9 `unsigned int Lumiere : :m_lumSoir` [protected]

valeur de la lumière du matin

Définition à la ligne 45 du fichier Lumiere.h.

5.8.4.10 `const int Lumiere : :m_maxCAD` [protected]

Définition à la ligne 54 du fichier Lumiere.h.

5.8.4.11 `bool Lumiere : :m_ouverture` [protected]

Définition à la ligne 46 du fichier Lumiere.h.

5.8.4.12 `const float Lumiere : :m_rapportConversion` [protected]

Définition à la ligne 53 du fichier Lumiere.h.

5.8.4.13 `const byte Lumiere : :m_tempsLum` [protected]

Définition à la ligne 56 du fichier Lumiere.h.

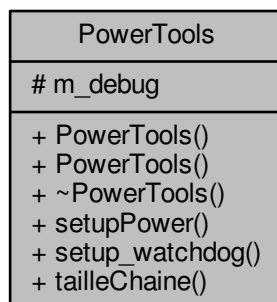
La documentation de cette classe a été générée à partir des fichiers suivants :

- /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/**Lumiere.**↔  
**h**
- /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/**Lumiere.**↔  
**cpp**

## 5.9 Référence de la classe PowerTools

```
#include <PowerTools.h>
```

Graphe de collaboration de PowerTools :



## Fonctions membres publiques

- **PowerTools** ()
- **PowerTools** (const boolean **debug**=false)
- **~PowerTools** ()
- void **setupPower** ()
  - initialisation power—
- void **setup\_watchdog** (int parametreTemps)
  - initialisation du watchdog - paramètre : 0=16ms, 1=32ms, 2=64ms, 3=128ms, 4=250ms, 5=500ms, 6=1 sec, 7=2 sec, 8=4 sec, 9=8 secondes—
- byte **tailleChaine** (char \*chaine)
  - taille d'une chaine de caractères—

## Attributs protégés

- const boolean **m\_debug**

### 5.9.1 Description détaillée

**PowerTools.h** (p. ??) prototype de la classe Power

- Gestion de la consommation en courant
- Watchdog
- Memory free
- outils

Définition à la ligne 18 du fichier PowerTools.h.

### 5.9.2 Documentation des constructeurs et destructeur

#### 5.9.2.1 **PowerTools** : **PowerTools** ( )

**PowerTools.h** (p. ??) définitions de la classe PowerToos

- Gestion de la consommation en courant
- Watchdog
- Memory free
- outils

Définition à la ligne 12 du fichier PowerTools.cpp.

#### 5.9.2.2 **PowerTools** : **PowerTools** ( const boolean *debug* = false )

Définition à la ligne 18 du fichier PowerTools.cpp.

#### 5.9.2.3 **PowerTools** : **~PowerTools** ( )

Définition à la ligne 23 du fichier PowerTools.cpp.

### 5.9.3 Documentation des fonctions membres

#### 5.9.3.1 void PowerTools : :setup\_watchdog ( int *parametreTemps* )

---initialisation du watchdog - paramètre : 0=16ms, 1=32ms, 2=64ms, 3=128ms, 4=250ms, 5=500ms, 6=1 sec, 7=2 sec, 8=4 sec, 9=8 secondes---

Définition à la ligne 43 du fichier PowerTools.cpp.

Voici le graphe des appelants de cette fonction :



#### 5.9.3.2 void PowerTools : :setupPower ( )

---initialisation power---

Définition à la ligne 29 du fichier PowerTools.cpp.

Voici le graphe des appelants de cette fonction :



#### 5.9.3.3 byte PowerTools : :tailleChaine ( char \* *chaine* )

---taille d'une chaine de caractères---

Définition à la ligne 61 du fichier PowerTools.cpp.

### 5.9.4 Documentation des données membres

#### 5.9.4.1 `const boolean PowerTools : :m_debug` [protected]

Définition à la ligne 36 du fichier PowerTools.h.

La documentation de cette classe a été générée à partir des fichiers suivants :

- /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/**PowerTools.h**
- /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/**PowerTools.cpp**

## 5.10 Référence de la classe Radio

```
#include <Radio.h>
```

Graphe de collaboration de Radio :

Radio
<pre># m_debug # m_pinEmRadio # m_pinSwitchEmissionRadio # m_vitesseTransmission # m_radio # m_taille # m_chaine</pre>
<pre>+ Radio() + ~Radio() + init() + raz_m_chaine() + envoiMessage() + messageRadio() + messageSansParametre() + chaineVide() + envoiFloat() + envoiUnsignedInt() + testSwitchEmissionRadio() + get_m_radio() + set_m_radio()</pre>



## Fonctions membres publiques

- **Radio** (byte **pinEmRadio**, byte **pinSwitchEmissionRadio**, int **vitesseTransmission**, byte *taille*, boolean **radio**=false, boolean **debug**=false)
- **~Radio** ()
- void **init** ()
  - initialisation—
- void **raz\_m\_chaine** ()
  - raz de m\_chaine—
- void **envoiMessage** (char \*chaine1)
  - routine construction message radio—
- void **messageRadio** (char \*chaine1)
  - message Radio—
- void **messageSansParametre** ()
  - message **Radio** (p. ??) sans parametre—
- void **chaineVide** ()
  - chaine radio fin de ligne avant transmission—
- void **envoiFloat** (float valeur, boolean **boitierOuvert**=true, char \*texte="")
  - envoi message float avec test de l'ouverture du boitier plus texte—
- void **envoiUnsignedInt** (unsigned int valeur, boolean **boitierOuvert**=true, char \*texte="")
  - envoi message unsigned int avec test de l'ouverture du boitier plus texte—
- void **testSwitchEmissionRadio** ()
  - test du switch emission radio on/off—
- bool **get\_m\_radio** ()
- void **set\_m\_radio** (bool radio)

## Attributs protégés

- const boolean **m\_debug**
- byte **m\_pinEmRadio**
- byte **m\_pinSwitchEmissionRadio**
- int **m\_vitesseTransmission**
- bool **m\_radio**
- int const **m\_taille**
- char **m\_chaine** []

## 5.10.1 Description détaillée

**Radio.h** (p. ??) prototype de la classe **Radio** (p. ??) radio 433MHz

Définition à la ligne 13 du fichier Radio.h.

## 5.10.2 Documentation des constructeurs et destructeur

5.10.2.1 **Radio** : **Radio** ( byte *pinEmRadio*, byte *pinSwitchEmissionRadio*, int *vitesseTransmission*, byte *taille*, boolean *radio* = false, boolean *debug* = false )

**Radio.cpp** (p. ??) définitions de la classe **Radio** (p. ??) radio 433MHz

Définition à la ligne 9 du fichier Radio.cpp.

5.10.2.2 **Radio** : **~Radio** ( )

Définition à la ligne 15 du fichier Radio.cpp.

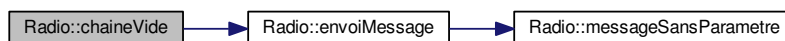
### 5.10.3 Documentation des fonctions membres

#### 5.10.3.1 void Radio::chaineVide ( )

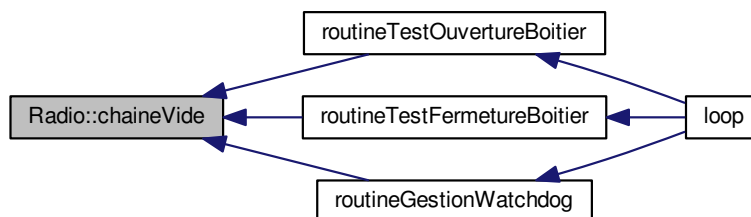
---chaine radio fin de ligne avant transmission---

Définition à la ligne 71 du fichier Radio.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

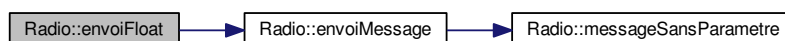


#### 5.10.3.2 void Radio::envoiFloat ( float valeur, boolean boitierOuvert = true, char \* texte = " " )

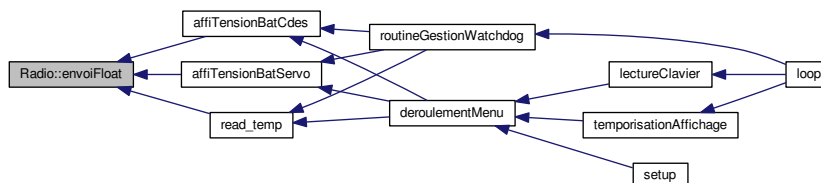
---envoi message float avec test de l'ouverture du boitier plus texte---

Définition à la ligne 80 du fichier Radio.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



### 5.10.3.3 void Radio::envoiMessage ( char \* *chaine1* )

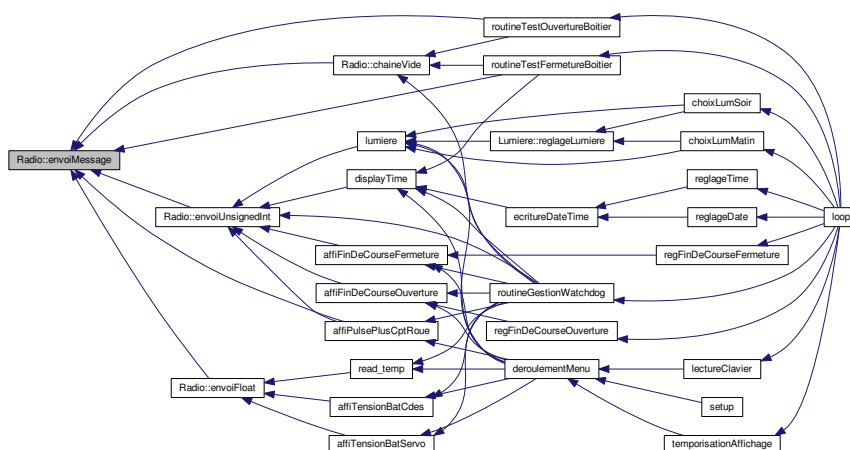
—routine construction message radio—

Définition à la ligne 53 du fichier Radio.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

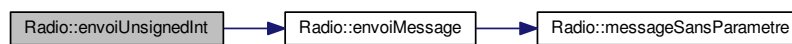


#### 5.10.3.4 void Radio : :envoiUnsignedInt ( unsigned int *valeur*, boolean *boitierOuvert* = true, char \* *texte* = " " )

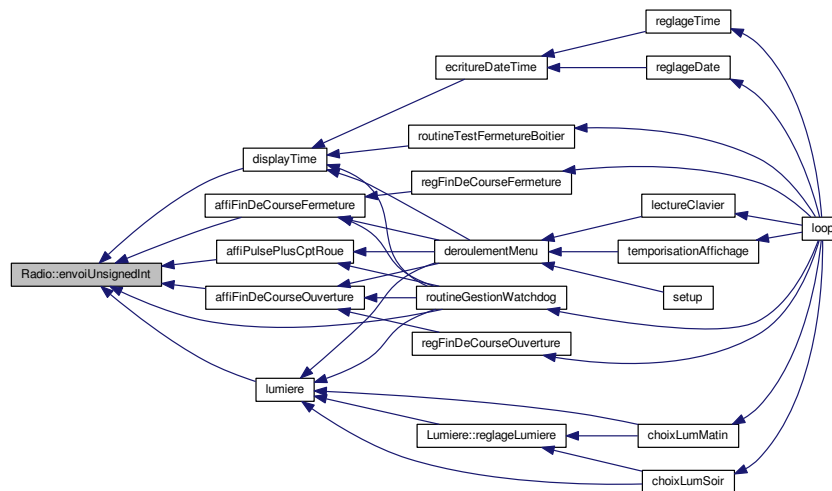
---envoi message unsigned int avec test de l'ouverture du boitier plus texte---

Définition à la ligne 92 du fichier Radio.cpp.

Voici le graphe d'appel pour cette fonction :



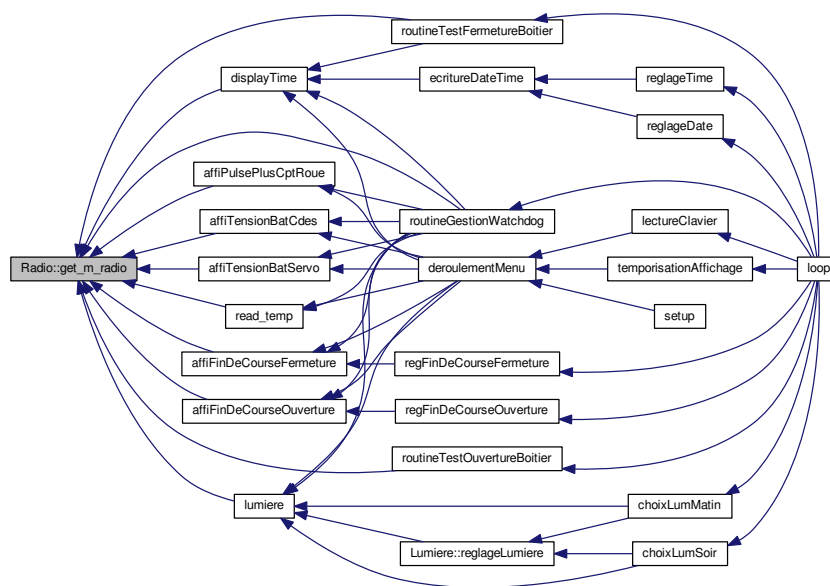
Voici le graphe des appelants de cette fonction :



#### 5.10.3.5 bool Radio : :get\_m\_radio ( )

Définition à la ligne 116 du fichier Radio.cpp.

Voici le graphe des appelants de cette fonction :



#### 5.10.3.6 void Radio::init ( )

---initialisation---

Définition à la ligne 20 du fichier Radio.cpp.

Voici le graphe des appelants de cette fonction :

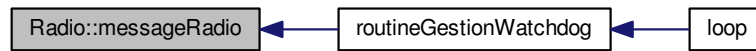


#### 5.10.3.7 void Radio::messageRadio ( char \* chaine1 )

---message Radio---

Définition à la ligne 32 du fichier Radio.cpp.

Voici le graphe des appelants de cette fonction :

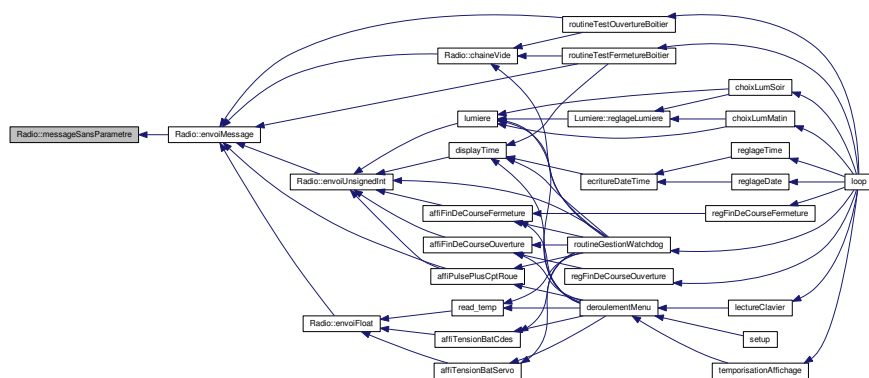


#### 5.10.3.8 void Radio::messageSansParametre ( )

—message **Radio** (p. ??) sans parametre—

Définition à la ligne 41 du fichier Radio.cpp.

Voici le graphe des appelants de cette fonction :

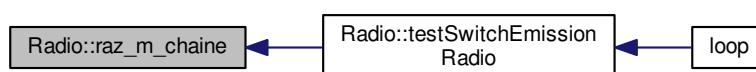


#### 5.10.3.9 void Radio::raz\_m\_chaine ( )

—raz de m\_chaine—

Définition à la ligne 27 du fichier Radio.cpp.

Voici le graphe des appelants de cette fonction :



5.10.3.10 `void Radio : :set_m_radio ( bool radio )`

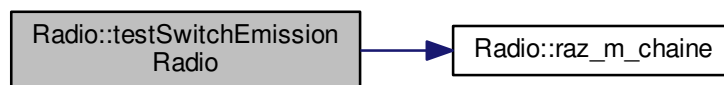
Définition à la ligne 121 du fichier Radio.cpp.

5.10.3.11 `void Radio : :testSwitchEmissionRadio ( )`

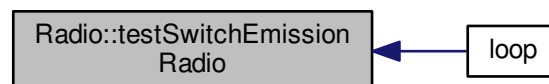
---test du switch emission radio on/off---

Définition à la ligne 104 du fichier Radio.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



## 5.10.4 Documentation des données membres

5.10.4.1 `char Radio : :m_chaine[]` [protected]

Définition à la ligne 43 du fichier Radio.h.

5.10.4.2 `const boolean Radio : :m_debug` [protected]

Définition à la ligne 37 du fichier Radio.h.

5.10.4.3 `byte Radio : :m_pinEmRadio` [protected]

Définition à la ligne 38 du fichier Radio.h.

#### 5.10.4.4 `byte Radio : :m_pinSwitchEmissionRadio` [protected]

Définition à la ligne 39 du fichier Radio.h.

#### 5.10.4.5 `bool Radio : :m_radio` [protected]

Définition à la ligne 41 du fichier Radio.h.

#### 5.10.4.6 `int const Radio : :m_taille` [protected]

Définition à la ligne 42 du fichier Radio.h.

#### 5.10.4.7 `int Radio : :m_vitesseTransmission` [protected]

Définition à la ligne 40 du fichier Radio.h.

La documentation de cette classe a été générée à partir des fichiers suivants :

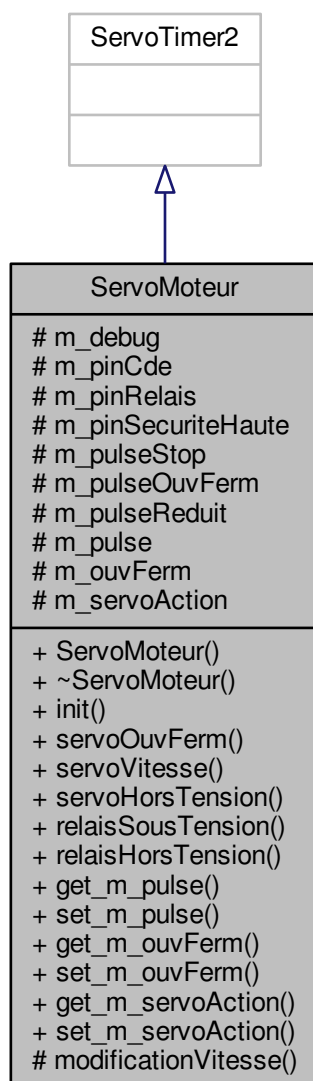
- /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/**Radio.h**
- /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/**Radio.**↔  
**cpp**

## 5.11 Référence de la classe ServoMoteur

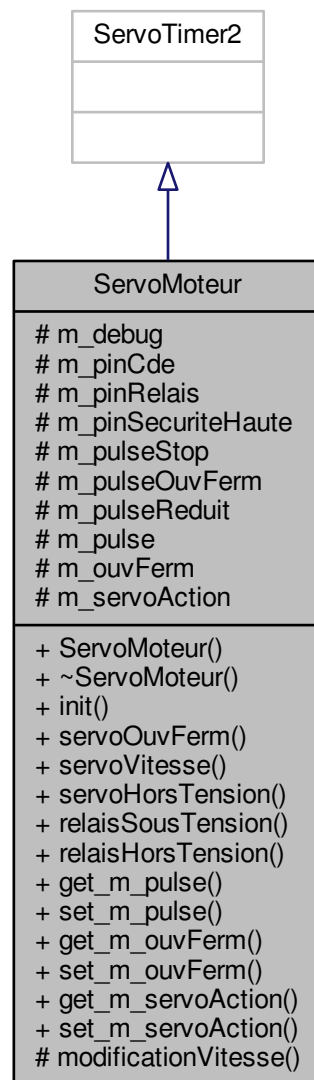
```
#include <ServoMoteur.h>
```



Graphe d'héritage de ServoMoteur :



Graphe de collaboration de ServoMoteur :



### Fonctions membres publiques

- **ServoMoteur** (const byte pinCde, const byte pinRelais, const byte pinSecuriteHaute, const int **pulse**, ←  
**Stop**=1500, const int pulseOuvFerm=140, const int pulseReduit=70, boolean **debug**=false)
- **~ServoMoteur** ()
- void **init** ()  
*——initialisation——*
- void **servoOuvFerm** (bool **batterieFaible**, bool **reduit**)  
*——mise sous tension du servo et ouverture fermeture de la porte——*
- void **servoVitesse** (bool **reduit**)  
*——modification de la vitesse si le servo en action——*
- unsigned int **servoHorsTension** (unsigned int **compteRoueCodeuse**, unsigned int finCourseOuverture)  
*——mise hors tension relais du servo——*
- void **relaisSousTension** ()

```

    ---relais sous tension---
— void relaisHorsTension ()
    ---relais hors tension---
— int get_m_pulse ()
— void set_m_pulse (int pulse)
— bool get_m_ouvFerm ()
— void set_m_ouvFerm (bool ouvFerm)
— bool get_m_servoAction ()
— void set_m_servoAction (bool servoAction)

```

### Fonctions membres protégées

```

— void modificationVitesse (bool reduit)
    ---modification de la vitesse---

```

### Attributs protégés

```

— const boolean m_debug
— const byte m_pinCde
— const byte m_pinRelais
— const byte m_pinSecuriteHaute
— const int m_pulseStop
— const int m_pulseOuvFerm
— const int m_pulseReduit
— int m_pulse
— bool m_ouvFerm
— bool m_servoAction

```

#### 5.11.1 Description détaillée

**ServoMoteur.h** (p. ??) prototype de la classe **ServoMoteur** (p. ??) gestion du servo moteur au travers de la bibliothèque ServoTimer2

Définition à la ligne 12 du fichier ServoMoteur.h.

#### 5.11.2 Documentation des constructeurs et destructeur

5.11.2.1 **ServoMoteur** : **ServoMoteur** ( const byte *pinCde*, const byte *pinRelais*, const byte *pinSecuriteHaute*, const int *pulseStop* = 1500, const int *pulseOuvFerm* = 140, const int *pulseReduit* = 70, boolean *debug* = false )

**ServoMoteur.cpp** (p. ??) définitions de la classe **ServoMoteur** (p. ??) gestion du servo moteur au travers de la bibliothèque ServoTimer2

Définition à la ligne 9 du fichier ServoMoteur.cpp.

5.11.2.2 **ServoMoteur** : **~ServoMoteur** ( )

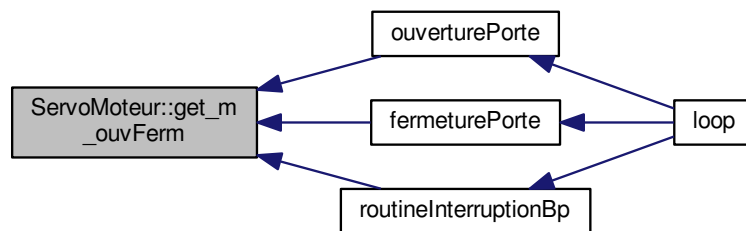
Définition à la ligne 16 du fichier ServoMoteur.cpp.

### 5.11.3 Documentation des fonctions membres

#### 5.11.3.1 `bool ServoMoteur : :get_m_ouvFerm ( )`

Définition à la ligne 93 du fichier ServoMoteur.cpp.

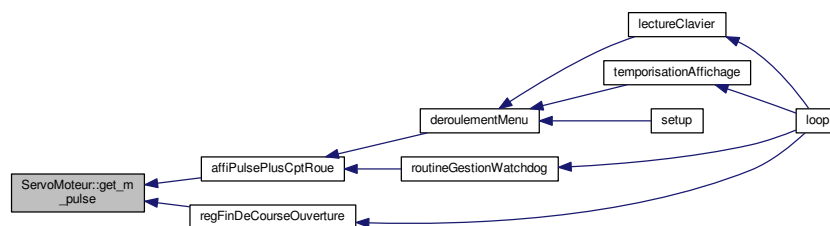
Voici le graphe des appelants de cette fonction :



#### 5.11.3.2 `int ServoMoteur : :get_m_pulse ( )`

Définition à la ligne 83 du fichier ServoMoteur.cpp.

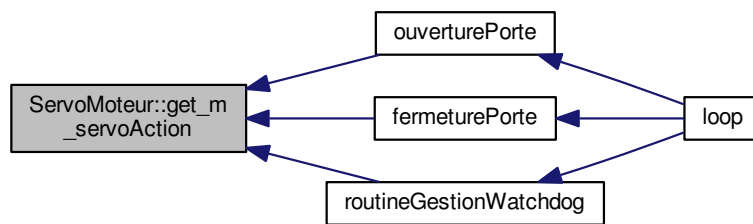
Voici le graphe des appelants de cette fonction :



#### 5.11.3.3 `bool ServoMoteur : :get_m_servoAction ( )`

Définition à la ligne 103 du fichier ServoMoteur.cpp.

Voici le graphe des appelants de cette fonction :

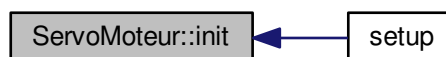


#### 5.11.3.4 void ServoMoteur : :init ( )

---initialisation---

Définition à la ligne 21 du fichier `ServoMoteur.cpp`.

Voici le graphe des appelants de cette fonction :

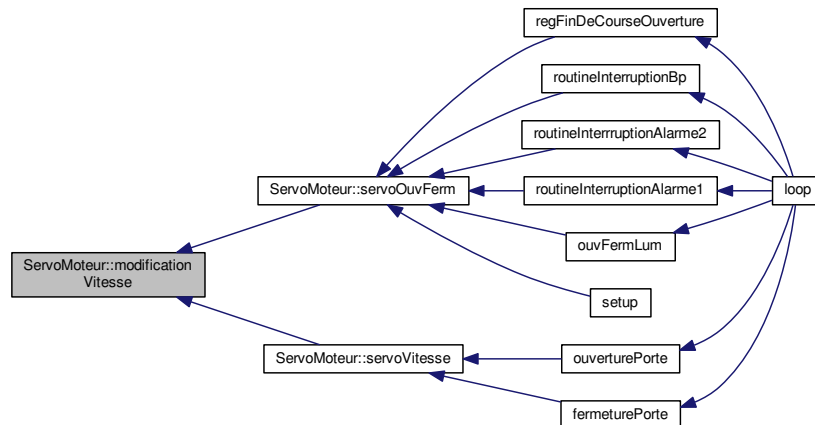


#### 5.11.3.5 void ServoMoteur : :modificationVitesse ( bool reduit ) [protected]

---modification de la vitesse---

Définition à la ligne 52 du fichier `ServoMoteur.cpp`.

Voici le graphe des appelants de cette fonction :

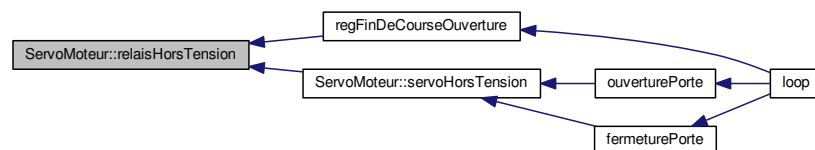


#### 5.11.3.6 void ServoMoteur : :relaisHorsTension ( )

---relais hors tension---

Définition à la ligne 72 du fichier ServoMoteur.cpp.

Voici le graphe des appelants de cette fonction :

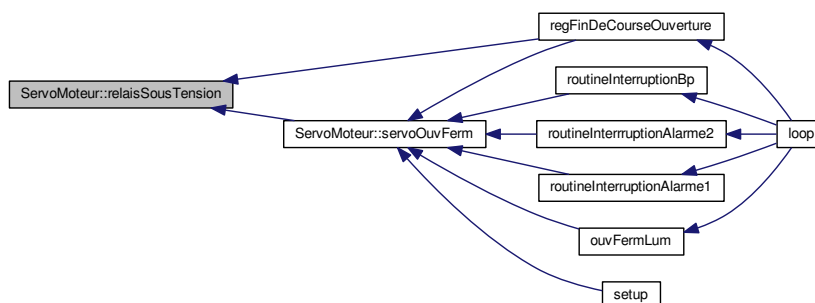


#### 5.11.3.7 void ServoMoteur : :relaisSousTension ( )

---relais sous tension---

Définition à la ligne 78 du fichier ServoMoteur.cpp.

Voici le graphe des appelants de cette fonction :



5.11.3.8 `unsigned int ServoMoteur::servoHorsTension ( unsigned int compteRoueCodeuse, unsigned int finCourseOuverture )`

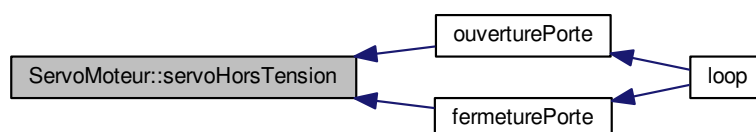
---mise hors tension relais du servo---

Définition à la ligne 59 du fichier `ServoMoteur.cpp`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

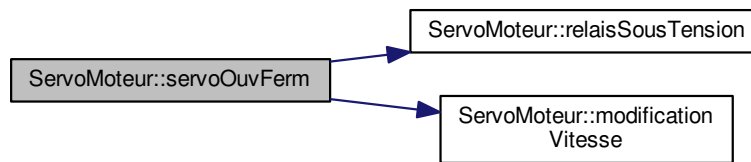


#### 5.11.3.9 void ServoMoteur : :servoOuvFerm ( bool *batterieFaible*, bool *reduit* )

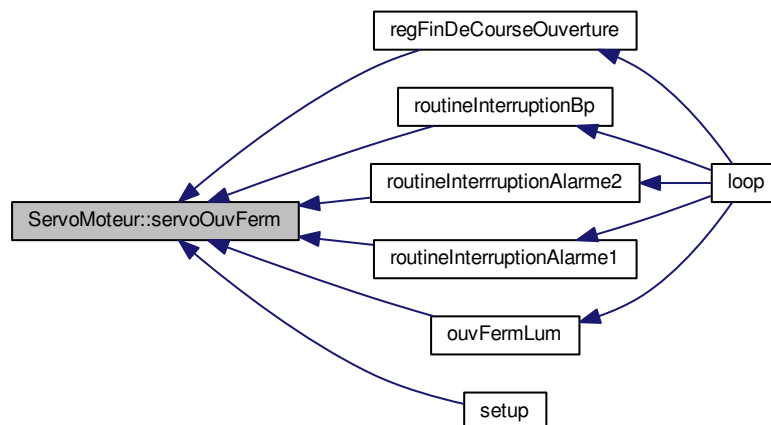
-----mise sous tension du servo et ouverture fermeture de la porte-----

Définition à la ligne 32 du fichier ServoMoteur.cpp.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



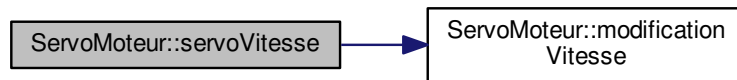
#### 5.11.3.10 void ServoMoteur : :servoVitesse ( bool *reduit* )

-----modificaton de la vitesse si le servo en action-----

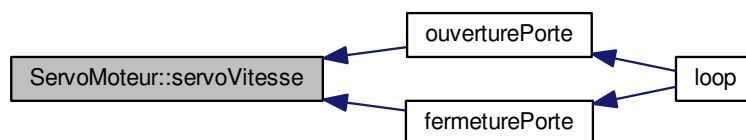
Définition à la ligne 44 du fichier ServoMoteur.cpp.



Voici le graphe d'appel pour cette fonction :



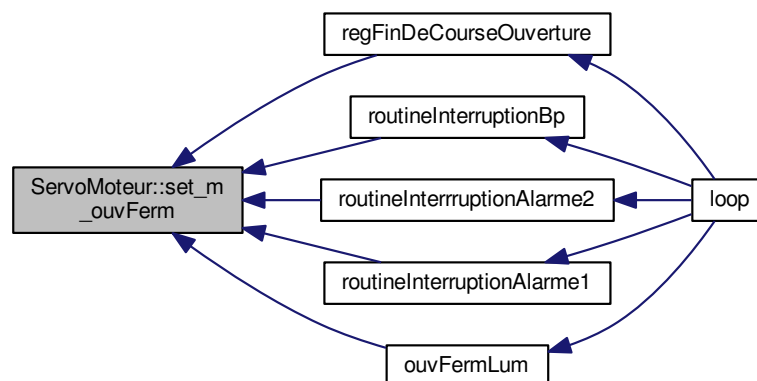
Voici le graphe des appelants de cette fonction :



#### 5.11.3.11 void ServoMoteur : :set\_m\_ouvFerm ( bool ouvFerm )

Définition à la ligne 98 du fichier ServoMoteur.cpp.

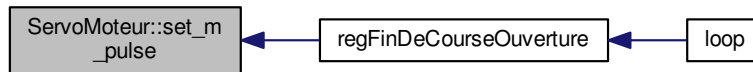
Voici le graphe des appelants de cette fonction :



#### 5.11.3.12 void ServoMoteur : :set\_m\_pulse ( int *pulse* )

Définition à la ligne 88 du fichier ServoMoteur.cpp.

Voici le graphe des appelants de cette fonction :



#### 5.11.3.13 void ServoMoteur : :set\_m\_servoAction ( bool *servoAction* )

Définition à la ligne 108 du fichier ServoMoteur.cpp.

### 5.11.4 Documentation des données membres

#### 5.11.4.1 const boolean ServoMoteur : :m\_debug [protected]

Définition à la ligne 42 du fichier ServoMoteur.h.

#### 5.11.4.2 bool ServoMoteur : :m\_ouvFerm [protected]

Définition à la ligne 50 du fichier ServoMoteur.h.

#### 5.11.4.3 const byte ServoMoteur : :m\_pinCde [protected]

Définition à la ligne 43 du fichier ServoMoteur.h.

#### 5.11.4.4 const byte ServoMoteur : :m\_pinRelais [protected]

Définition à la ligne 44 du fichier ServoMoteur.h.

#### 5.11.4.5 const byte ServoMoteur : :m\_pinSecuriteHaute [protected]

Définition à la ligne 45 du fichier ServoMoteur.h.

#### 5.11.4.6 int ServoMoteur : :m\_pulse [protected]

Définition à la ligne 49 du fichier ServoMoteur.h.

5.11.4.7 `const int ServoMoteur : :m_pulseOuvFerm` [protected]

Définition à la ligne 47 du fichier ServoMoteur.h.

5.11.4.8 `const int ServoMoteur : :m_pulseReduit` [protected]

Définition à la ligne 48 du fichier ServoMoteur.h.

5.11.4.9 `const int ServoMoteur : :m_pulseStop` [protected]

Définition à la ligne 46 du fichier ServoMoteur.h.

5.11.4.10 `bool ServoMoteur : :m_servoAction` [protected]

Définition à la ligne 51 du fichier ServoMoteur.h.

La documentation de cette classe a été générée à partir des fichiers suivants :

- /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/**ServoMoteur.h**
- /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/**ServoMoteur.cpp**



## Chapitre 6

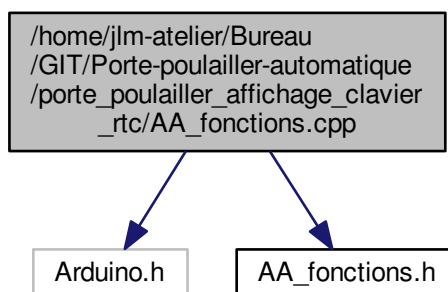
# Documentation des fichiers

6.1 Référence du fichier `/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/carte  
heuristique porte poulailler.mm`

6.2 Référence du fichier `/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_↵  
_poulailler_affichage_clavier_rtc/AA_fonctions.cpp`

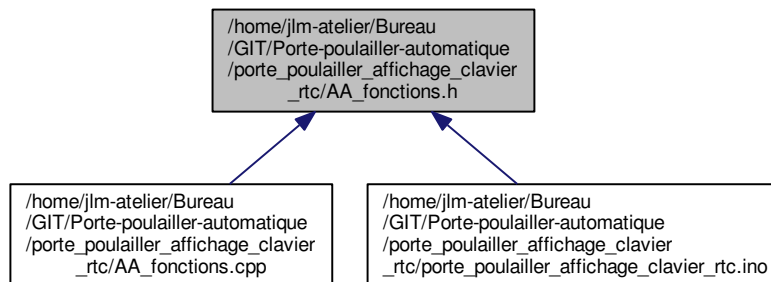
```
#include "Arduino.h"  
#include "AA_fonctions.h"
```

Graphe des dépendances par inclusion de `AA_fonctions.cpp` :



### 6.3 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/AA\_fonctions.h

Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



### Fonctions

- void **lectureClavier** ()  
    —lecture clavier—
- void **temporisationAffichage** (const int **boucleTemps**)  
    — temporisation pour l'affichage : affichage du menu lorsque temps est > boucleTemps—
- void **ecritureDateTime** ()  
    —routine lecture et ecriture date and time—
- void **displayDate** ()  
    —routine display Date—
- void **displayTime** ()  
    —routine display Time—
- void **openTime** ()  
    —routine door open time—
- void **closeTime** ()  
    —routine door close time—
- void **affiPulsePlusCptRoue** ()  
    —affichage pulse et comptage roue codeuse—
- void **eclairageAfficheur** ()  
    —retro eclaireage de l'afficheur—
- void **affiTensionBatCdes** ()  
    —affichage tension batterie commandes
- void **affiTensionBatServo** ()  
    —affichage tension batterie servo-moteur
- void **affiChoixOuvFerm** ()  
    —affichage du choix de l'ouverture et la fermeture—
- void **choixOuvFerm** ()  
    —routine choix ouverture fermeture—
- void **reglageHeureFermeture** ()  
    —reglage de l'heure de fermeture—
- void **reglageHeureOuverture** ()  
    —routine de reglage de l'heure d'ouverture—
- void **reglageDate** ()  
    —routine reglage jour semaine, jour, mois, annee—
- void **affiLumMatin** ()  
    —affichage de la lumiere du matin—
- void **affiLumSoir** ()  
    —affichage de la lumiere du soir—
- void **choixLumSoir** ()  
    —reglage du choix de la lumiere du soir—
- void **reglageTime** ()

```
-----routine reglage heure , minute , seconde-----
— void affiFinDeCourseFermeture ()
-----affichage fin de course Fermeture-----
— void affiFinDeCourseOuverture ()
-----affichage fin de course Ouverture-----
— void regFinDeCourseFermeture ()
-----reglage fin de course Fermeture-----
— void regFinDeCourseOuverture ()
-----reglage fin de course ouverture-----
— void testServo ()
    reglage du servo plus test de la roue codeuse et du servo, à l'aide de la console
— void read_temp (const boolean typeTemperature)
    -----routine lecture température sur ds3231 rtc type celsius=true , fahrenheit=false-----
— void ouverturePorte ()
    -----sequence ouverture de la porte-----
— void fermeturePorte ()
    -----sequence fermeture de la porte-----
— void myInterruptINT0 ()
    -----routine interruption D2 INT0-----
— void myInterruptINT1 ()
— void routineInterruptionBp ()
    -----routine interruption Bp-----
— void routineInterruptionAlarme2 ()
    -----routine alarme 2-----
— void routineInterruptionAlarme1 ()
    -----routine alarme 1-----
— void routineTestOuvertureBoitier ()
    -----test ouverture boitier-----
— void routineTestFermetureBoitier ()
    -----test fermeture boitier-----
— void lumiere ()
    -----routine lecture et affichage de la lumière-----
— void ouvFermLum ()
    -----ouverture/fermeture par test de la lumière-----
— void deroulementMenu (byte increment)
    -----routine affichage menus-----
— ISR (WDT_vect)
    -----Watchdog Interrupt Service est exécuté lors d'un timeout du WDT-----
— void enterSleep (void)
    -----entree dans le mode sleep-----
— void routineGestionWatchdog ()
    -----routine de gestion du watchdog-----
— void affichageDemarrage (byte colonne)
    -----routine affichage au demarrage-----
```

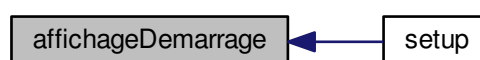
### 6.3.1 Documentation des fonctions

#### 6.3.1.1 void affichageDemarrage ( byte colonne )

-----routine affichage au demarrage-----

Définition à la ligne 967 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe des appelants de cette fonction :

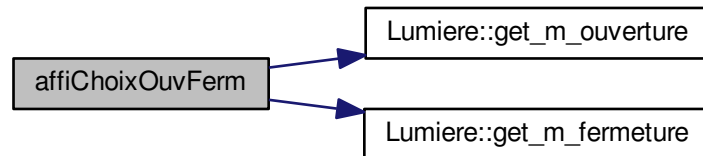


### 6.3.1.2 void affiChoixOuvFerm ( )

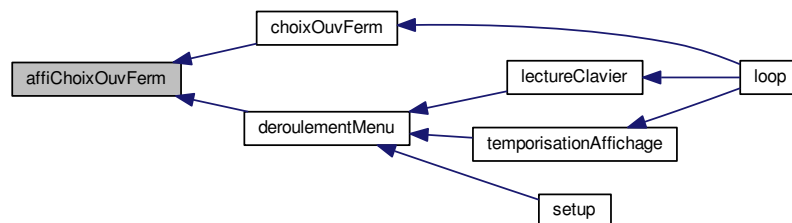
-----affichage du choix de l'ouverture et la fermeture-----

Définition à la ligne 338 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

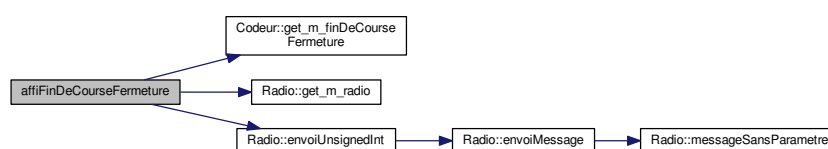


### 6.3.1.3 void affiFinDeCourseFermeture ( )

-----affichage fin de course Fermeture-----

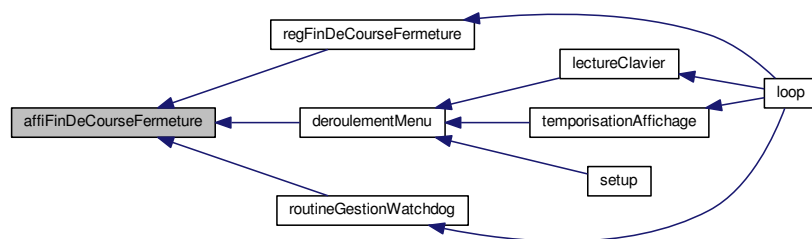
Définition à la ligne 534 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :





Voici le graphe des appelants de cette fonction :

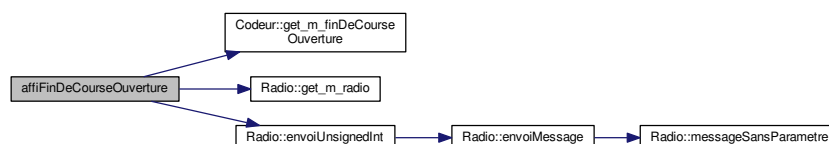


#### 6.3.1.4 void affiFinDeCourseOuverture ( )

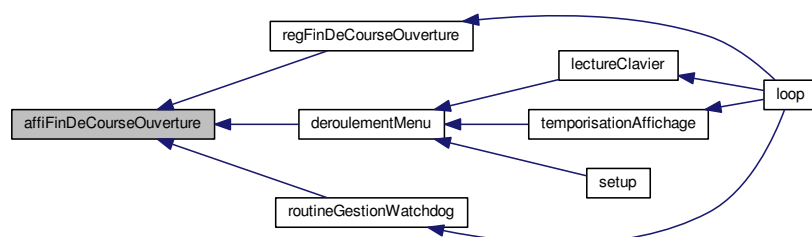
-----affichage fin de course Ouverture-----

Définition à la ligne 545 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



### 6.3.1.5 void affiLumMatin ( )

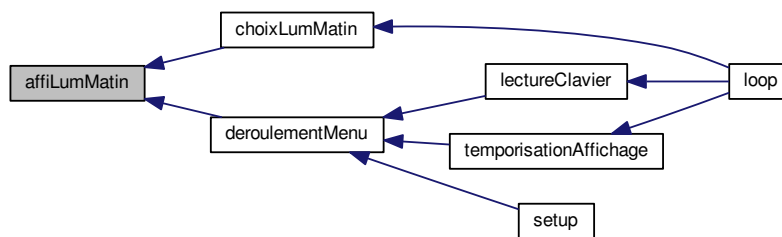
-----affichage de la lumiere du matin-----

Définition à la ligne 448 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



### 6.3.1.6 void affiLumSoir ( )

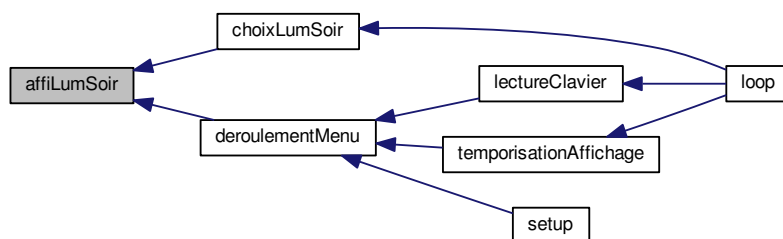
-----affichage de la lumiere du soir-----

Définition à la ligne 478 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

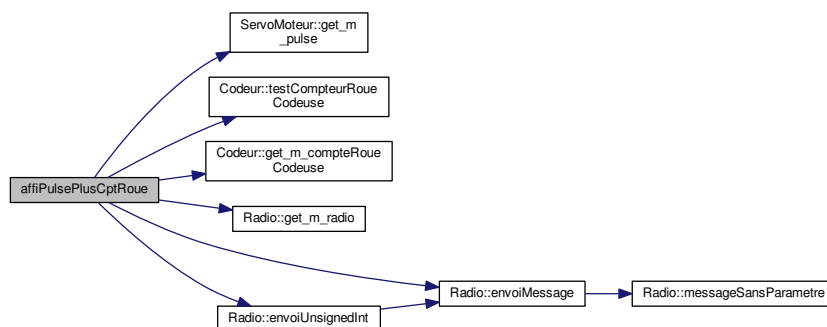


### 6.3.1.7 void affiPulsePlusCptRoue ( )

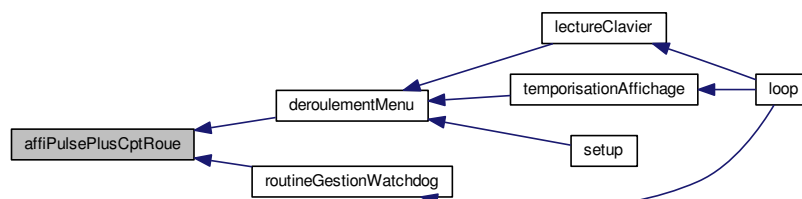
-----affichage pulse et comptage roue codeuse-----

Définition à la ligne 270 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

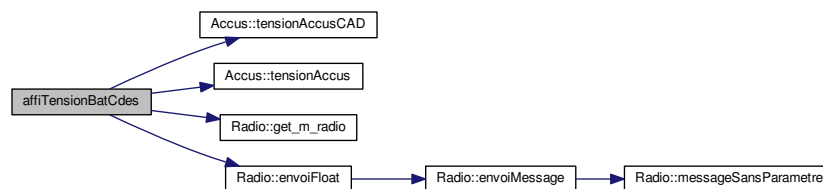


### 6.3.1.8 void affiTensionBatCdes ( )

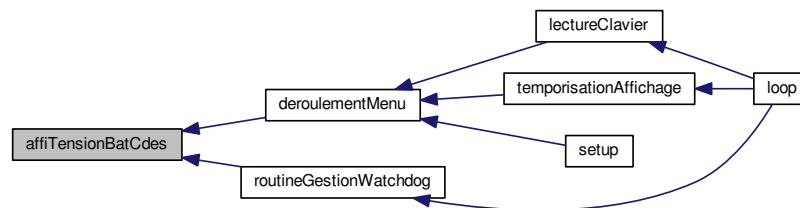
-----affichage tension batterie commandes

Définition à la ligne 308 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

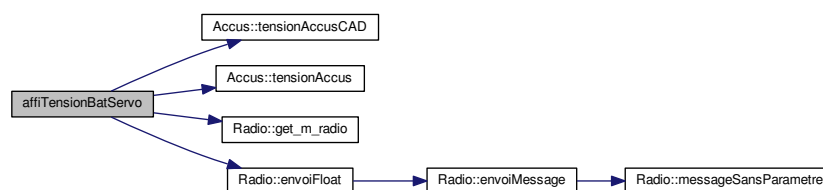


### 6.3.1.9 void affiTensionBatServo ( )

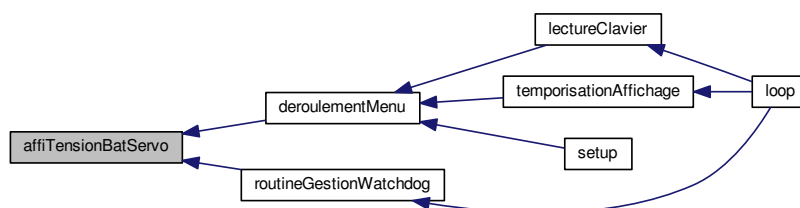
-----affichage tension batterie servo-moteur

Définition à la ligne 321 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

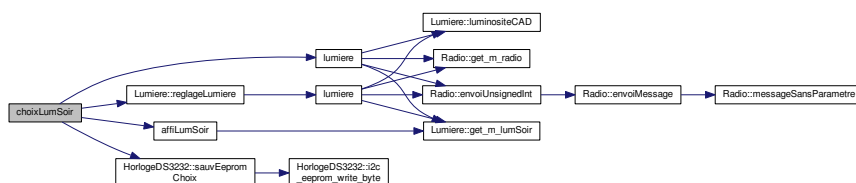


### 6.3.1.10 void choixLumSoir ( )

-----reglage du choix de la lumiere du soir-----

Définition à la ligne 487 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

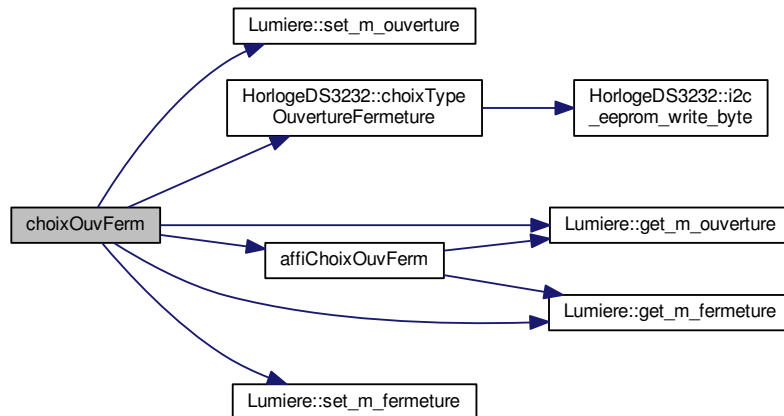


### 6.3.1.11 void choixOuvFerm ( )

----routine choix ouverture fermeture----

Définition à la ligne 346 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

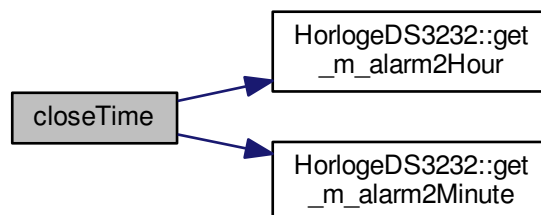


#### 6.3.1.12 void closeTime ( )

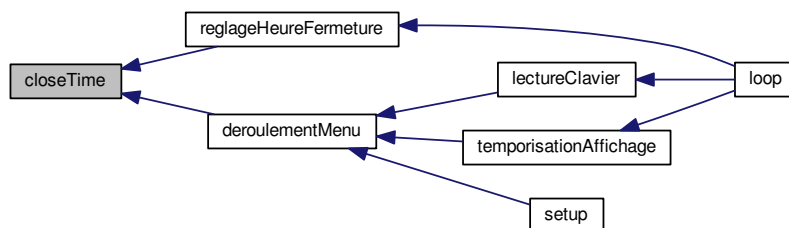
---routine door close time---

Définition à la ligne 262 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

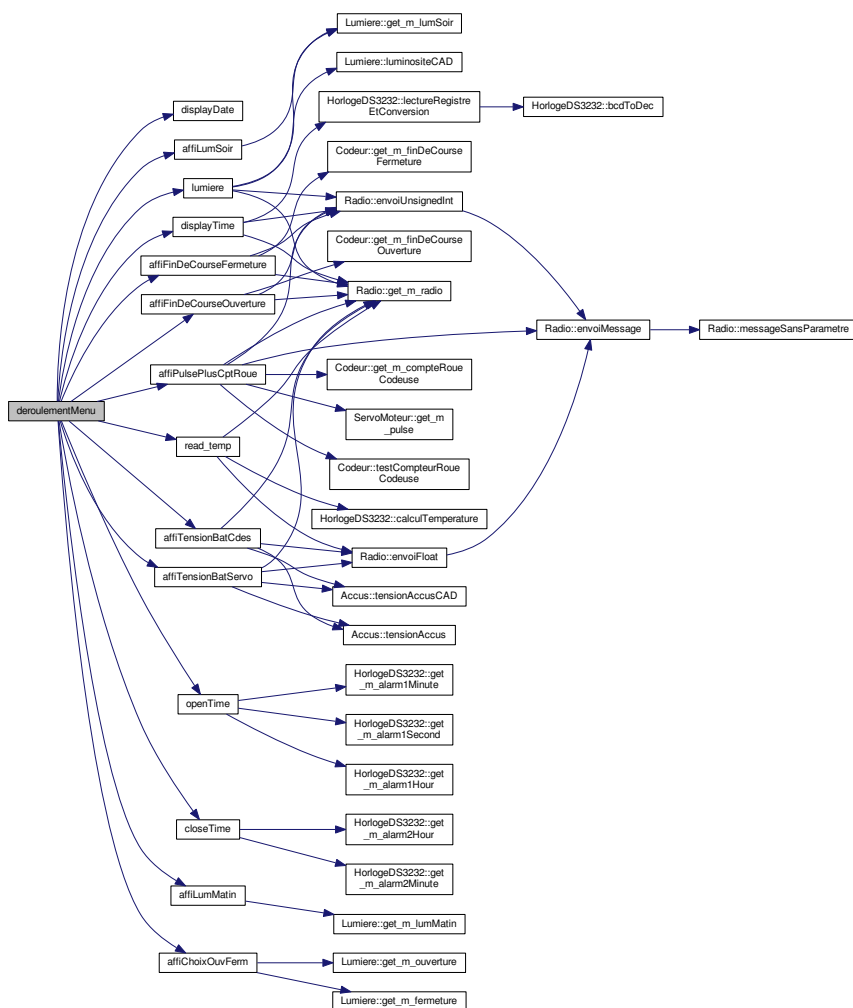


### 6.3.1.13 void deroulementMenu ( byte increment )

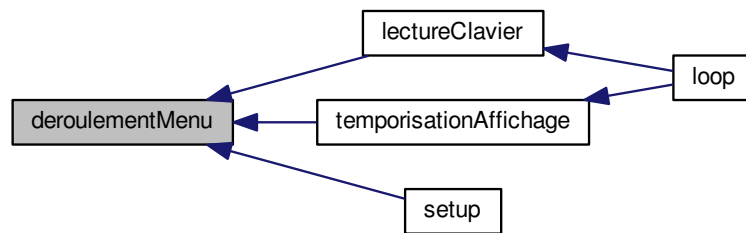
---routine affichage menus---

Définition à la ligne 855 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

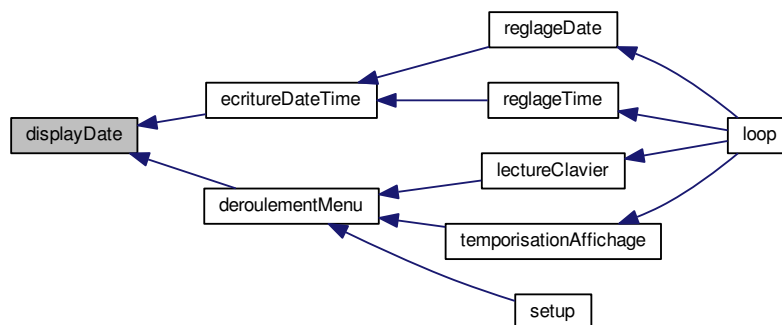


#### 6.3.1.14 void displayDate ( )

---routine display Date---

Définition à la ligne 227 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

Voici le graphe des appelants de cette fonction :



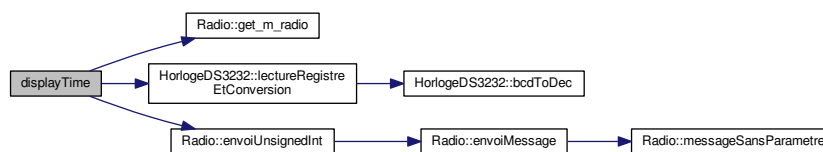
#### 6.3.1.15 void displayTime ( )

---routine display Time---

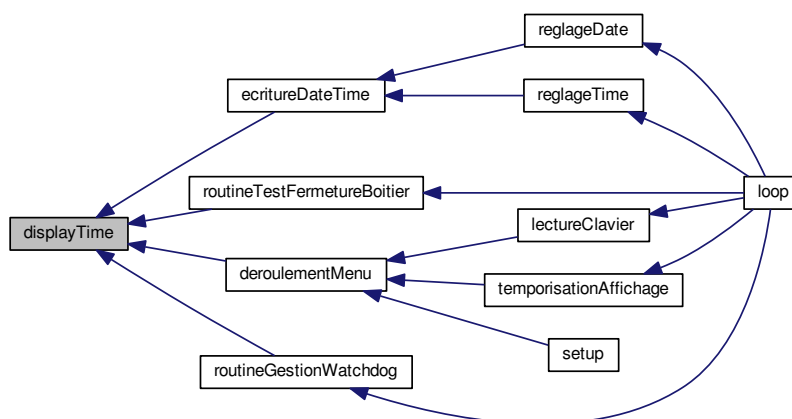
Définition à la ligne 241 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.



Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 6.3.1.16 void eclaireageAfficheur ( )

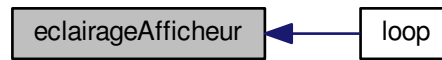
---retro eclaireage de l'afficheur---

Définition à la ligne 298 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

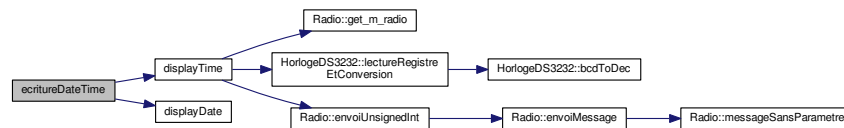


### 6.3.1.17 void ecritureDateTime ( )

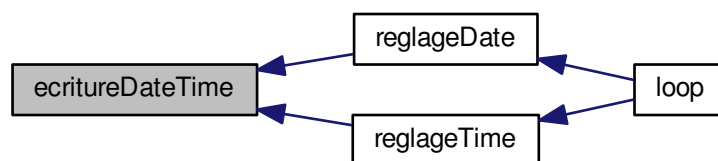
—routine lecture et ecriture date and time—

Définition à la ligne 215 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



### 6.3.1.18 void enterSleep ( void )

—entree dans le mode sleep—

Définition à la ligne 919 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe des appelants de cette fonction :

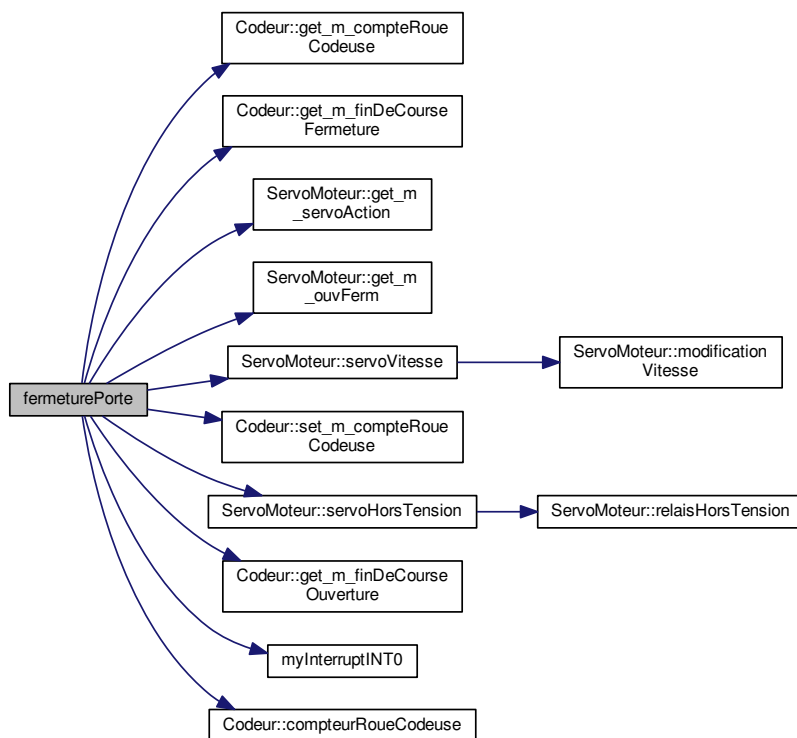


#### 6.3.1.19 void fermeturePorte ( )

---sequence fermeture de la porte---

Définition à la ligne 710 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 6.3.1.20 ISR ( WDT\_vect )

—Watchdog Interrupt Service est exécuté lors d'un timeout du WDT—

Définition à la ligne 912 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

#### 6.3.1.21 void lectureClavier ( )

—lecture clavier—

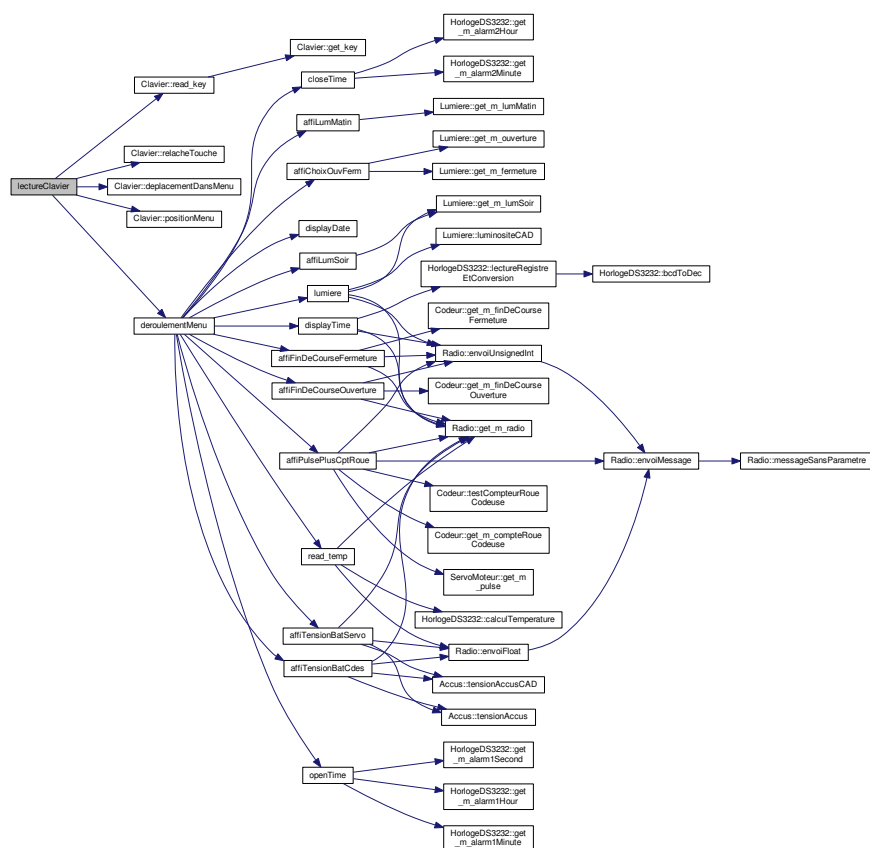
read key sensor = A1

position du menu pour l'affichage

affichage du menu

Définition à la ligne 193 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

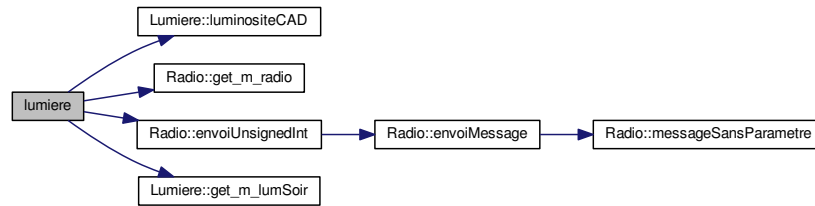


**6.3.1.22 void lumiere ( )**

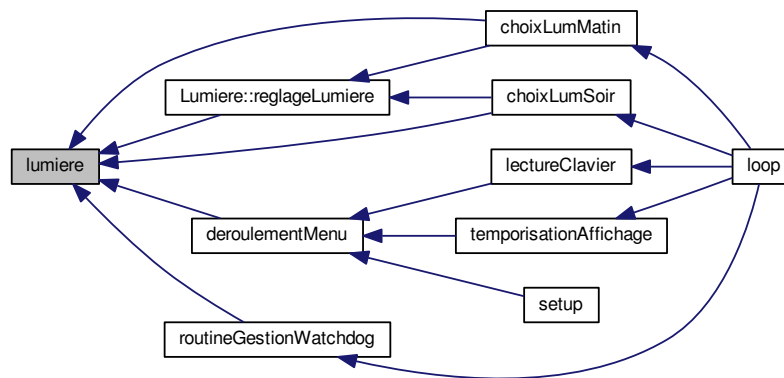
---routine lecture et affichage de la lumière---

Définition à la ligne 818 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



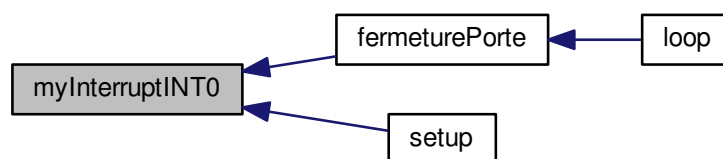
### 6.3.1.23 void myInterruptINT0 ( )

---routine interruption D2 INT0---

interruptions -routine interruption D2 INT0 -routine interruption D3 INT1 -routine interruption Bp -routine alarme 2  
-routine alarme 1 -routine interruption boitier ouvert -test fermeture boitier

Définition à la ligne 734 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

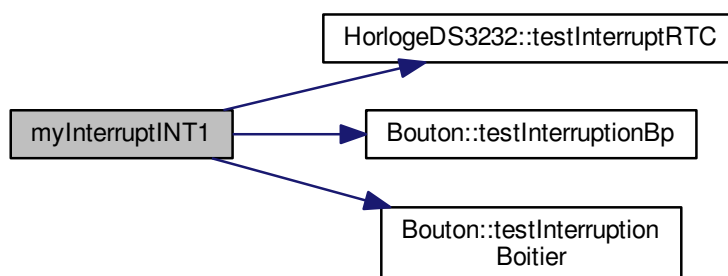
Voici le graphe des appelants de cette fonction :



## 6.3.1.24 void myInterruptINT1 ( )

Définition à la ligne 739 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

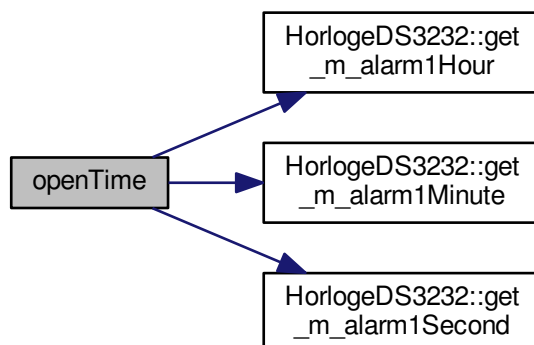


## 6.3.1.25 void openTime ( )

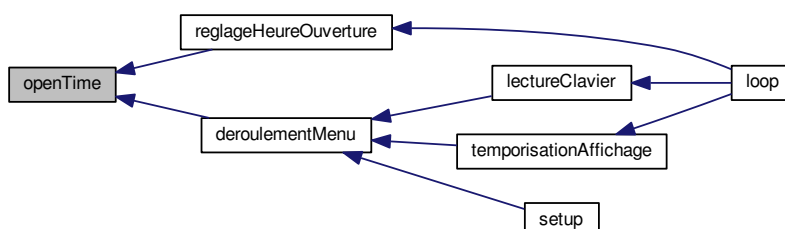
—routine door open time—

Définition à la ligne 254 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



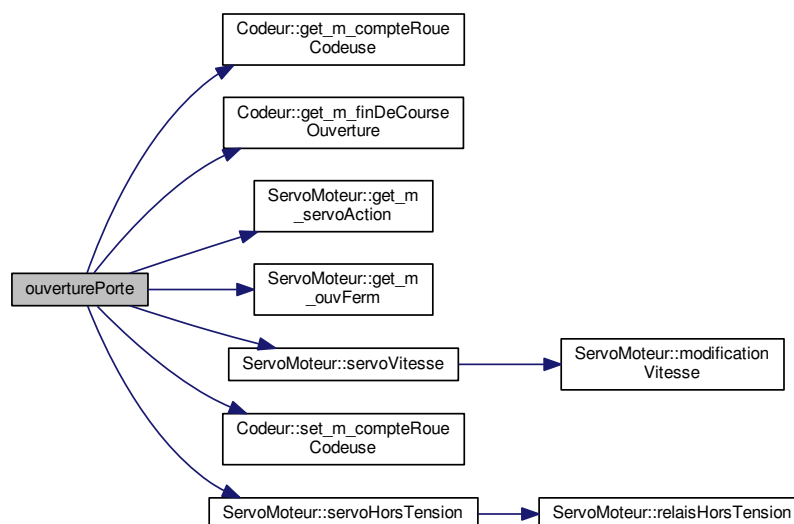
#### 6.3.1.26 void ouverturePorte ( )

-----sequence ouverture de la porte-----

Définition à la ligne 695 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.



Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

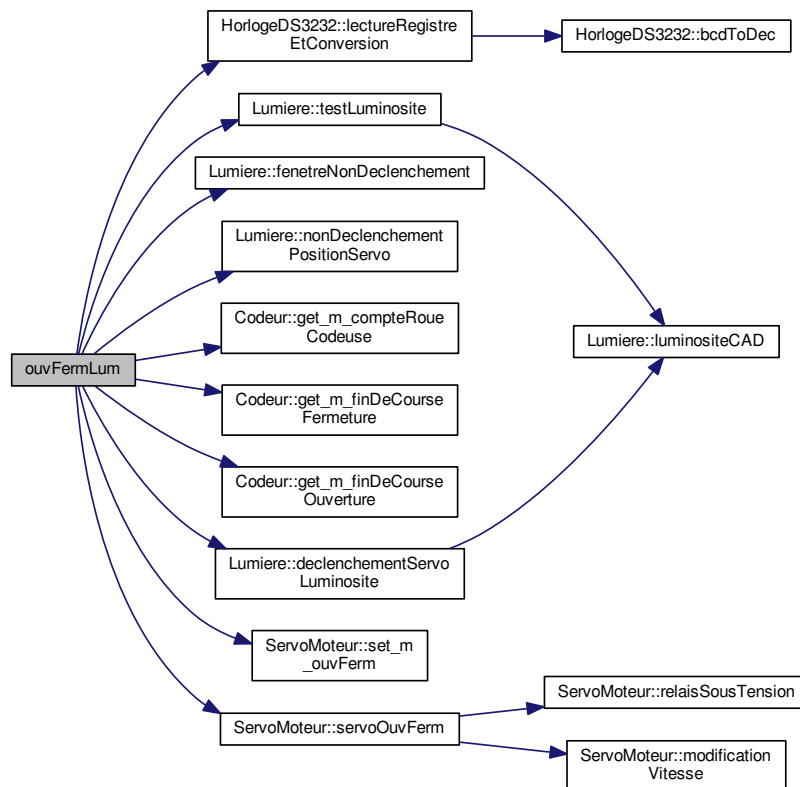


### 6.3.1.27 void ouvFermLum ( )

---ouverture/fermeture par test de la lumière---

Définition à la ligne 831 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

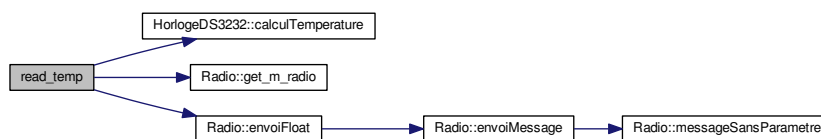


### 6.3.1.28 void read\_temp ( const boolean *typeTemperature* )

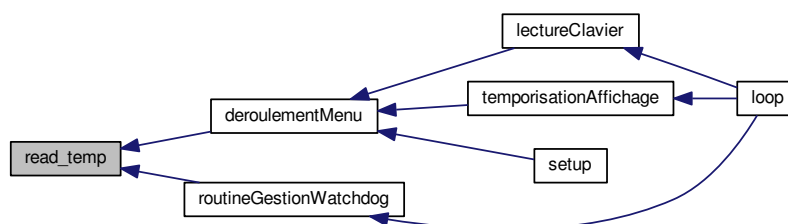
-----routine lecture température sur ds3231 rtc type celsius=true ,fahrenheit=false-----

Définition à la ligne 676 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

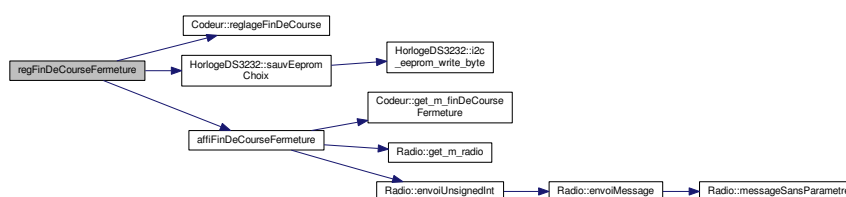


#### 6.3.1.29 void regFinDeCourseFermeture ( )

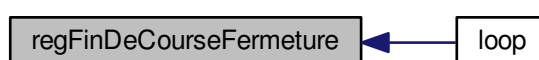
-----reglage fin de course Fermeture-----

Définition à la ligne 556 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

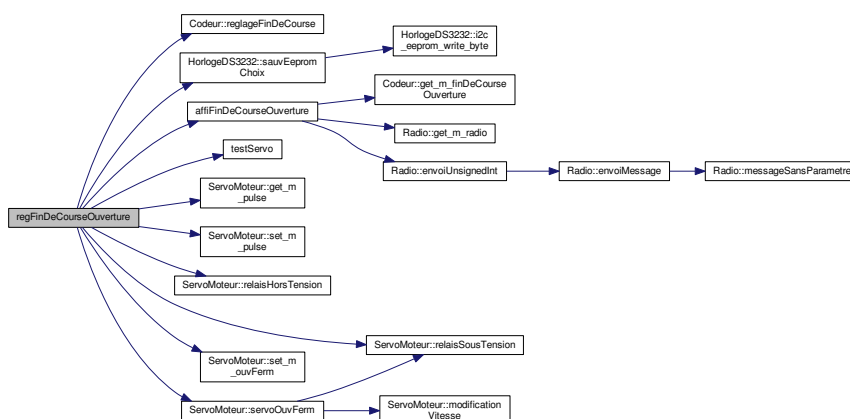


### 6.3.1.30 void regFinDeCourseOuverture ( )

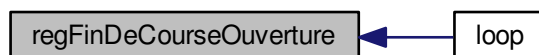
---regalge fin de course ouverture---

Définition à la ligne 576 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

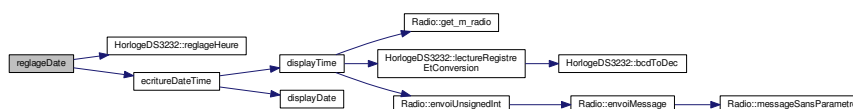


### 6.3.1.31 void reglageDate ( )

---routine reglage jour semaine, jour, mois, annee---

Définition à la ligne 418 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

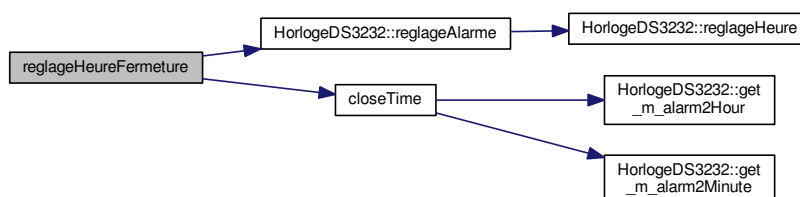


#### 6.3.1.32 void reglageHeureFermeture ( )

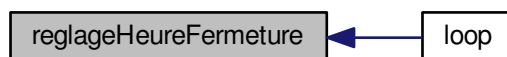
----reglage de l'heure de fermeture----

Définition à la ligne 368 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

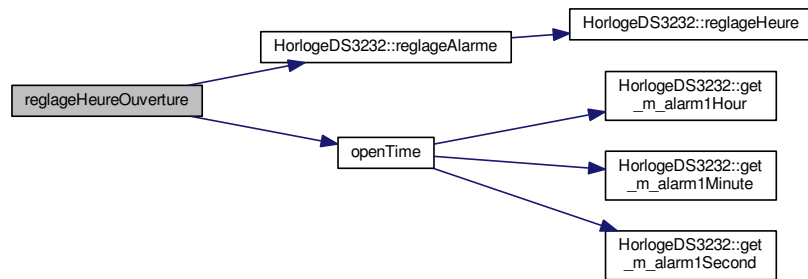


#### 6.3.1.33 void reglageHeureOuverture ( )

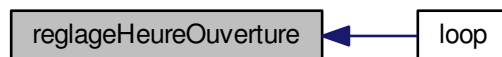
----routine de reglage de l'heure d'ouverture----

Définition à la ligne 391 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

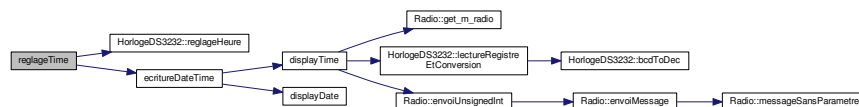


#### 6.3.1.34 void reglageTime ( )

---routine reglage heure , minute , seconde---

Définition à la ligne 508 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

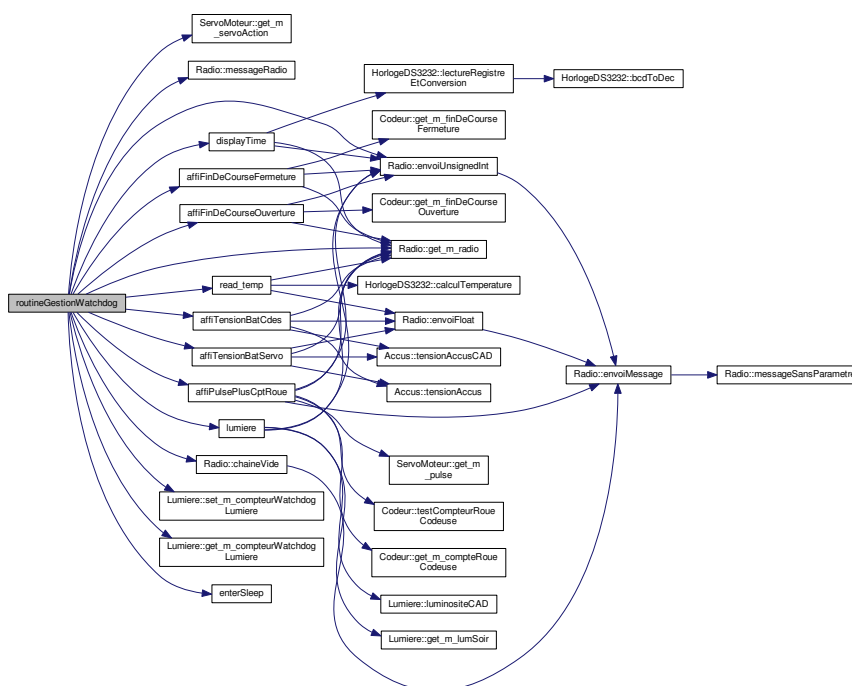


### 6.3.1.35 void routineGestionWatchdog ( )

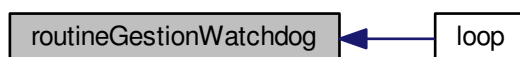
---routine de gestion du watchdog---

Définition à la ligne 928 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

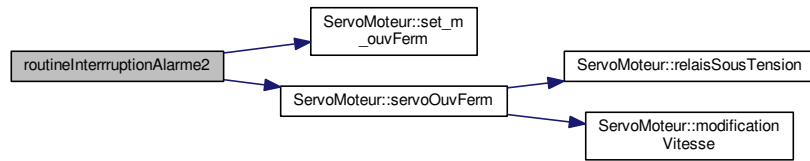


### 6.3.1.36 void routineInterruptionAlarme2 ( )

---routine alarme 2---

Définition à la ligne 758 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

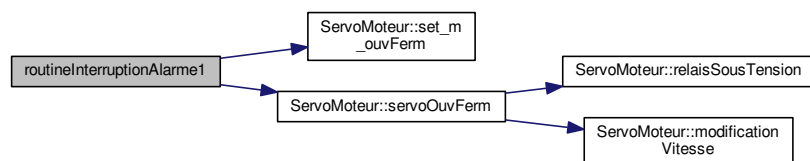


#### 6.3.1.37 void routineInterruptionAlarme1 ( )

---routine alarme 1---

Définition à la ligne 769 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



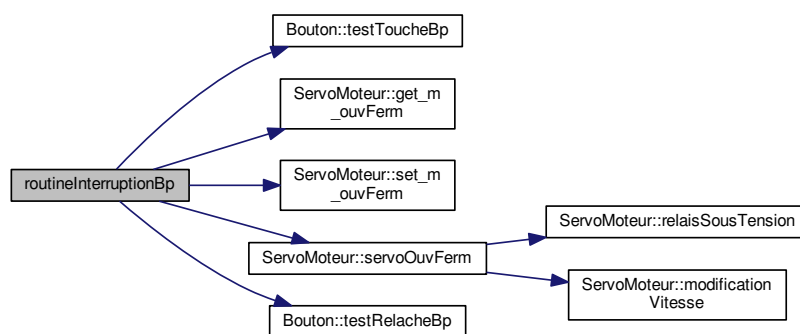


### 6.3.1.38 void routineInterruptionBp ( )

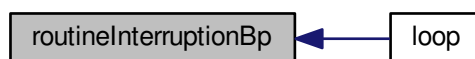
---routine interruption Bp---

Définition à la ligne 746 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

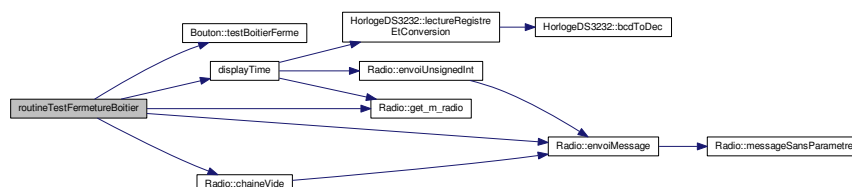


### 6.3.1.39 void routineTestFermetureBoitier ( )

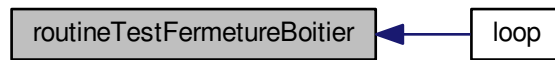
---test fermeture boitier---

Définition à la ligne 796 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

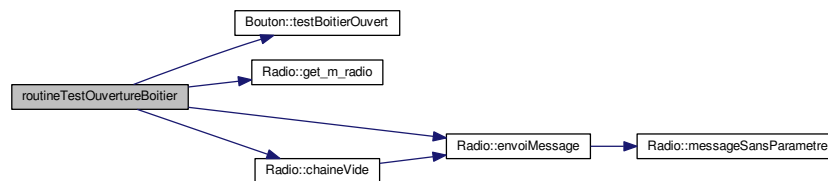


#### 6.3.1.40 void routineTestOuvertureBoitier ( )

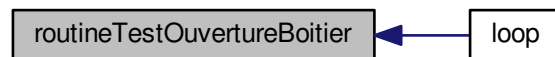
----test ouverture boitier----

Définition à la ligne 780 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

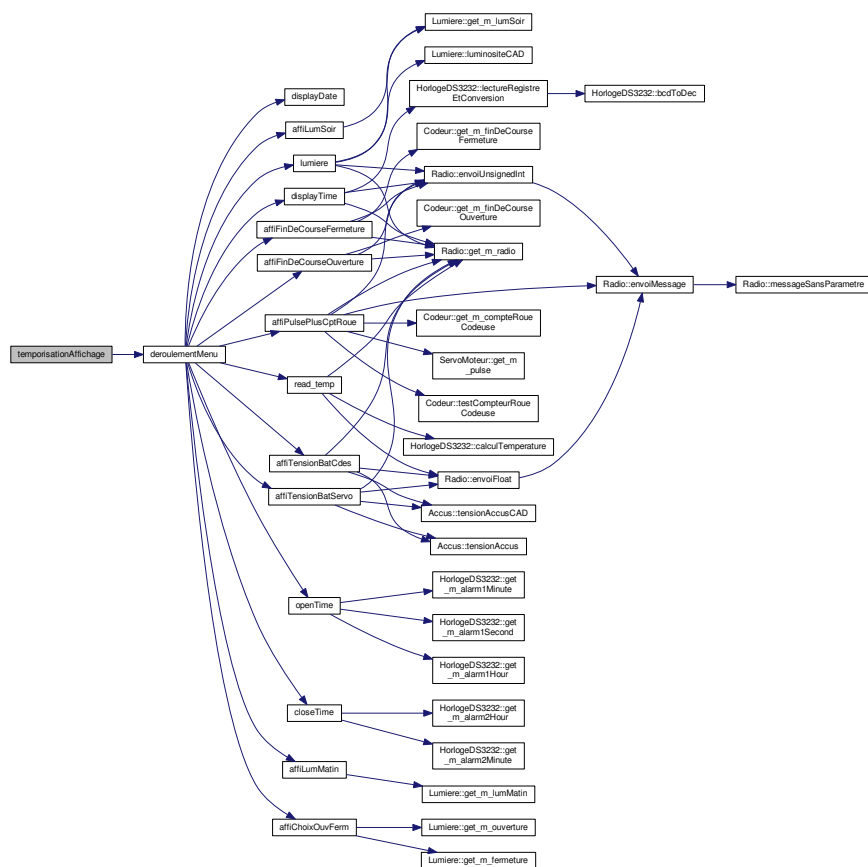


#### 6.3.1.41 void temporisationAffichage ( const int boucleTemps )

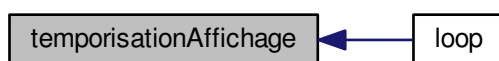
— temporisation pour l'affichage : affichage du menu lorsque temps est > boucleTemps—

Définition à la ligne 205 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 6.3.1.42 void testServo ( )

reglage du servo plus test de la roue codeuse et du servo, à l'aide de la console

Test du servo avec la console arduino

a = -1 q = -1

z = +10 s = -10

e = arrêt du servo (relais) d = marche du servo (relais)

r = montée f = descente

Définition à la ligne 611 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

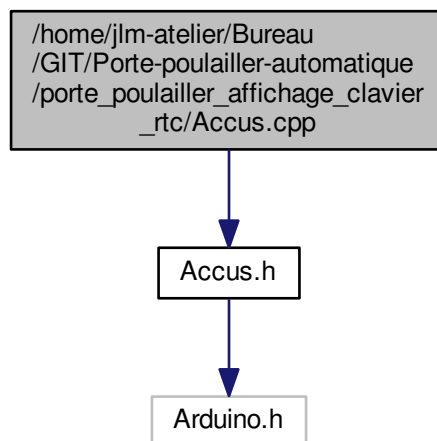
Voici le graphe des appelants de cette fonction :



## 6.4 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/Accus.cpp

```
#include "Accus.h"
```

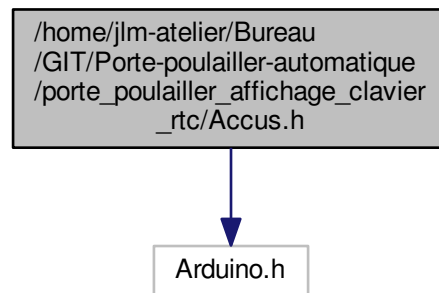
Grappe des dépendances par inclusion de Accus.cpp :



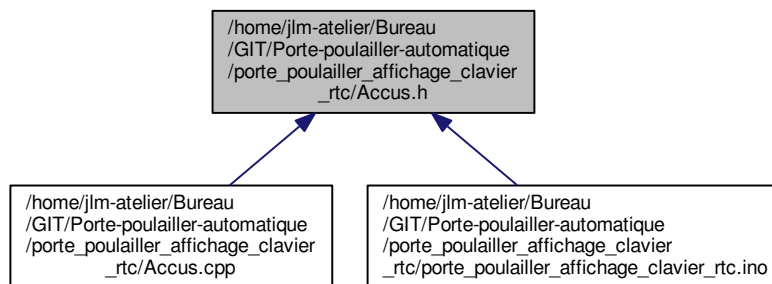
## 6.5 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/Accus.h

```
#include "Arduino.h"
```

Graphe des dépendances par inclusion de Accus.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

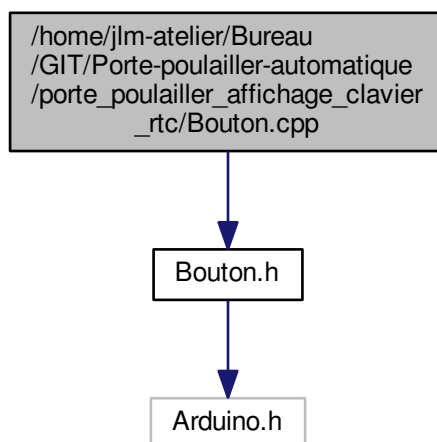
— class **Accus**

*Gestion des blocs accumulateurs NIMH de 4,8v.*

## 6.6 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_↔ \_poulailler\_affichage\_clavier\_rtc/Bouton.cpp

```
#include "Bouton.h"
```

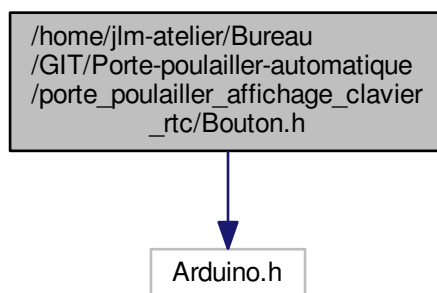
Graphe des dépendances par inclusion de Bouton.cpp :



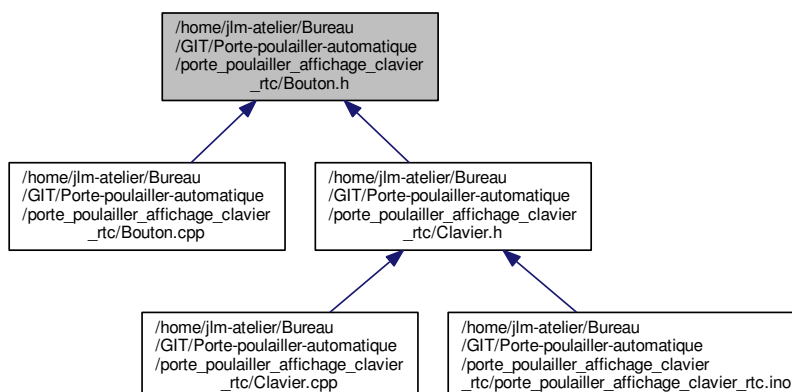
## 6.7 Référence du fichier `/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/Bouton.h`

```
#include "Arduino.h"
```

Graphe des dépendances par inclusion de Bouton.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



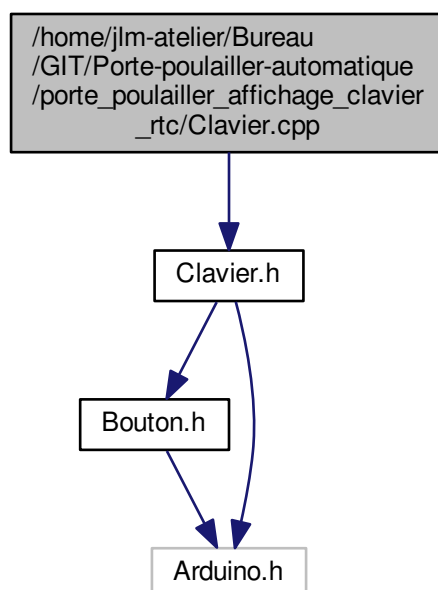
## Classes

— class **Bouton**

## 6.8 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_↵ affichage\_clavier\_rtc/Clavier.cpp

```
#include "Clavier.h"
```

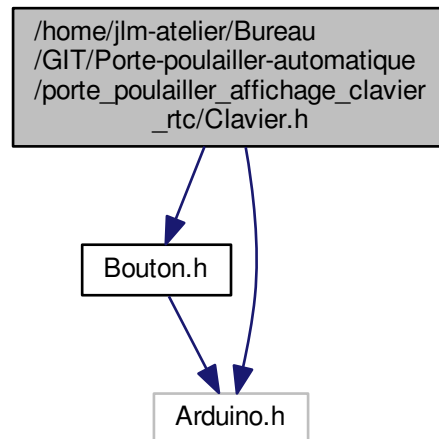
Grappe des dépendances par inclusion de Clavier.cpp :



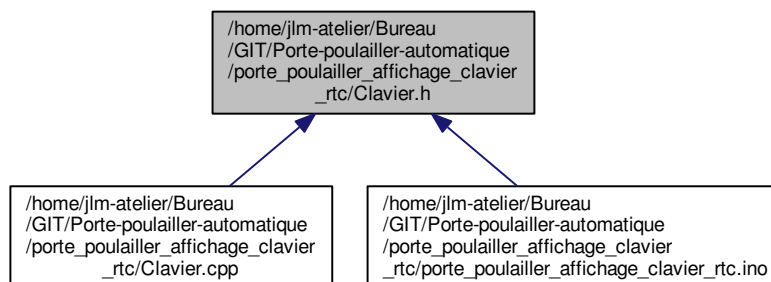
## 6.9 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/Clavier.h

```
#include "Bouton.h"  
#include "Arduino.h"
```

Graphe des dépendances par inclusion de Clavier.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

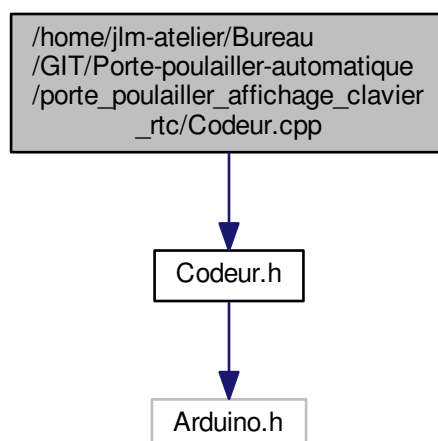
— class **Clavier**



## 6.10 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_↵ \_poulailler\_affichage\_clavier\_rtc/Codeur.cpp

```
#include "Codeur.h"
```

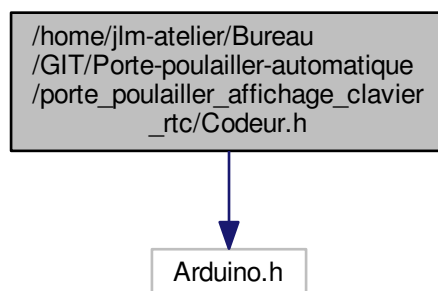
Graphe des dépendances par inclusion de Codeur.cpp :



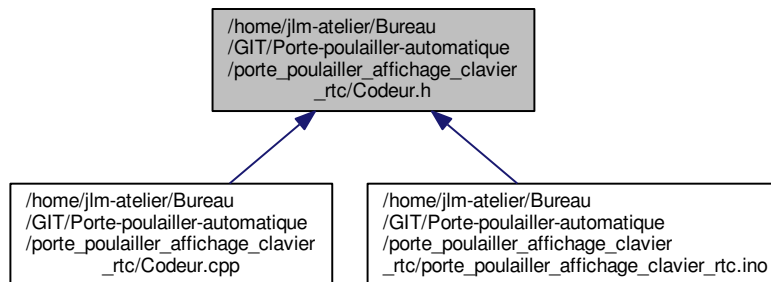
## 6.11 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_↵ \_poulailler\_affichage\_clavier\_rtc/Codeur.h

```
#include "Arduino.h"
```

Graphe des dépendances par inclusion de Codeur.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



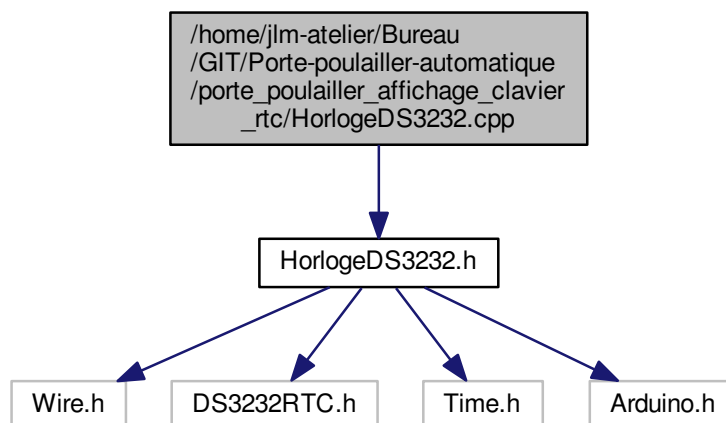
## Classes

— class **Codeur**

### 6.12 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/HorlogeDS3232.cpp

```
#include "HorlogeDS3232.h"
```

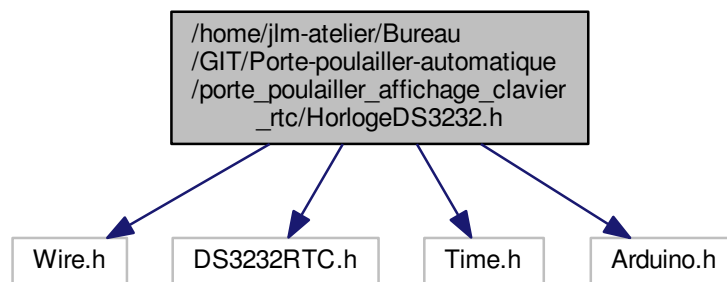
Graphe des dépendances par inclusion de `HorlogeDS3232.cpp` :



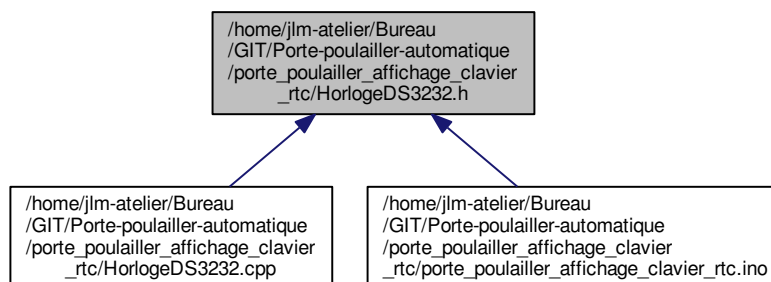
### 6.13 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_↵ \_poulailler\_affichage\_clavier\_rtc/HorlogeDS3232.h

```
#include <Wire.h>
#include <DS3232RTC.h>
#include <Time.h>
#include "Arduino.h"
```

Graphe des dépendances par inclusion de HorlogeDS3232.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

— class **HorlogeDS3232**

## Macros

— #define **DS3231\_I2C\_ADDRESS** 0x68

### 6.13.1 Documentation des macros

#### 6.13.1.1 `#define DS3231_I2C_ADDRESS 0x68`

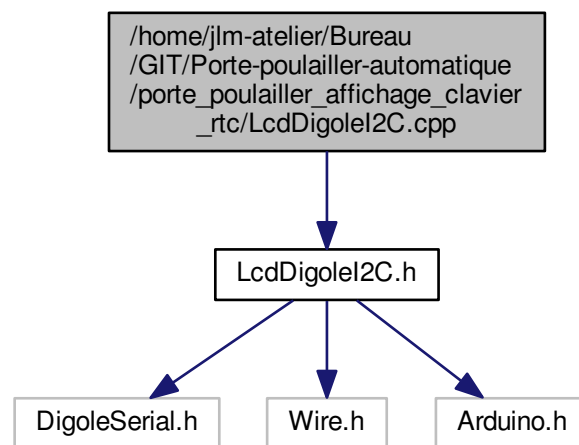
**HorlogeDS3232.h** (p. ??) prototype de la classe **HorlogeDS3232** (p. ??) RTC DS3232 avec température et circuit mémoire

Définition à la ligne 12 du fichier HorlogeDS3232.h.

## 6.14 Référence du fichier `/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/LcdDigoleI2C.cpp`

```
#include "LcdDigoleI2C.h"
```

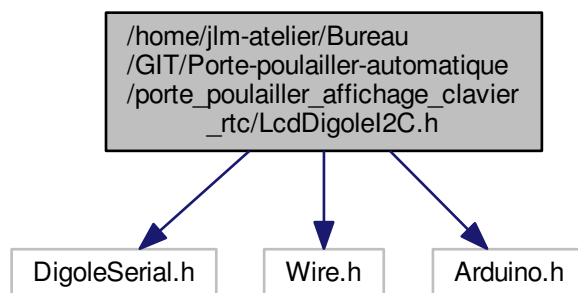
Graphe des dépendances par inclusion de LcdDigoleI2C.cpp :



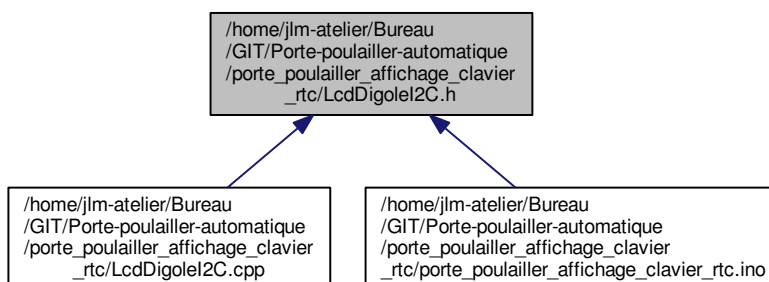
## 6.15 Référence du fichier `/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte_poulailler_affichage_clavier_rtc/LcdDigoleI2C.h`

```
#include <DigoleSerial.h>
#include <Wire.h>
#include "Arduino.h"
```

Graphe des dépendances par inclusion de LcdDigoleI2C.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



## Classes

— class **LcdDigoleI2C**

## Macros

— #define **\_Digole\_Serial\_I2C\_**

### 6.15.1 Documentation des macros

#### 6.15.1.1 #define \_Digole\_Serial\_I2C\_

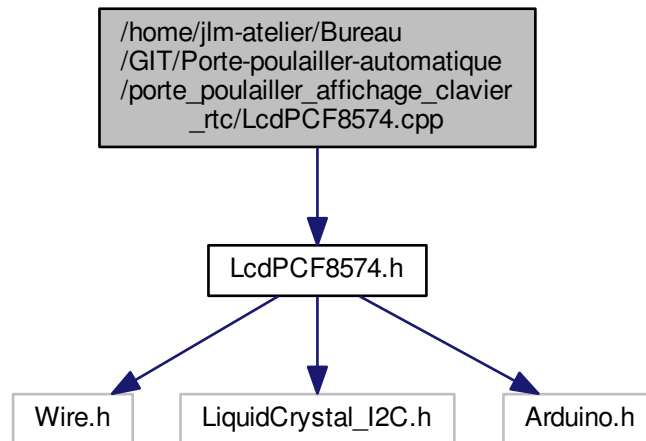
**LcdDigoleI2C.h** (p. ??) prototype de la classe LCD Digole I2C afficheur lcd 2\*16 caractères avec liaison serie i2c digoleSerial

Définition à la ligne 19 du fichier LcdDigoleI2C.h.

## 6.16 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/LcdPCF8574.cpp

```
#include "LcdPCF8574.h"
```

Graphe des dépendances par inclusion de LcdPCF8574.cpp :



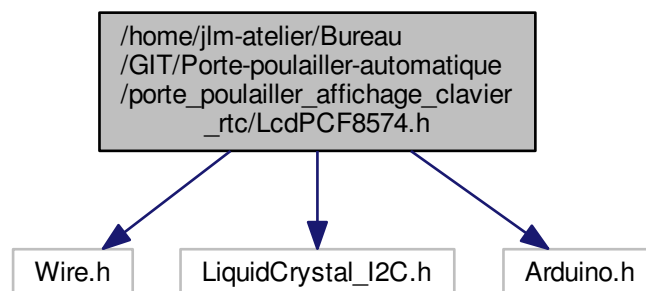
## 6.17 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/LcdPCF8574.h

```
#include <Wire.h>
```

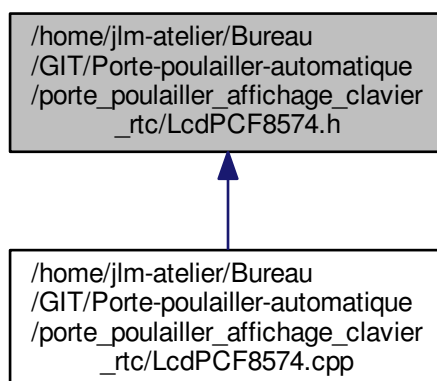
```
#include <LiquidCrystal_I2C.h>
```

```
#include "Arduino.h"
```

Graphe des dépendances par inclusion de LcdPCF8574.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



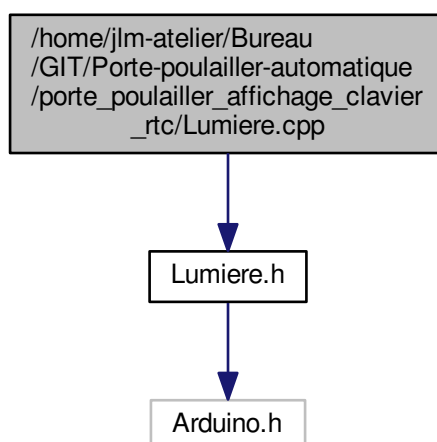
## Classes

— class **LcdPCF8574**

## 6.18 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_↵ \_poulailler\_affichage\_clavier\_rtc/Lumiere.cpp

```
#include "Lumiere.h"
```

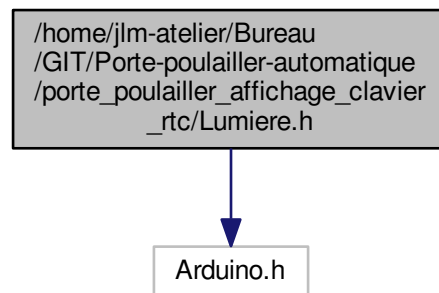
Graphe des dépendances par inclusion de Lumiere.cpp :



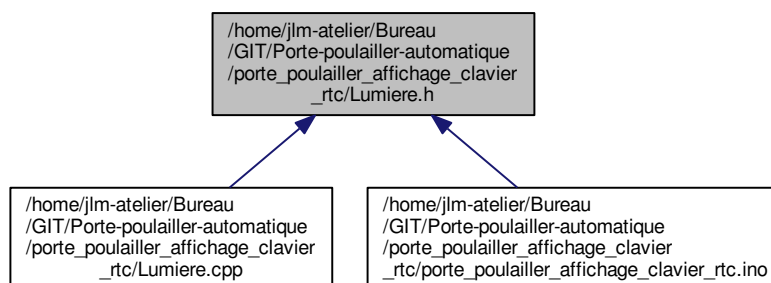
## 6.19 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/Lumiere.h

```
#include "Arduino.h"
```

Graphe des dépendances par inclusion de Lumiere.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



### Classes

— class **Lumiere**

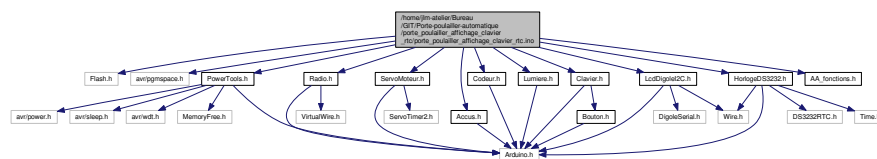
## 6.20 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/porte\_poulailler\_affichage\_clavier\_rtc.ino

```
#include <Flash.h>
```



```
#include <avr/pgmspace.h>
#include "PowerTools.h"
#include "Radio.h"
#include "ServoMoteur.h"
#include "Accus.h"
#include "Codeur.h"
#include "Lumiere.h"
#include "Clavier.h"
#include "LcdDigoleI2C.h"
#include "HorlogeDS3232.h"
#include "AA_fonctions.h"
```

Graphes des dépendances par inclusion de porte\_poulailler\_affichage\_clavier\_rtc.ino :



## Macros

- `#define LCD_DIGOLE`  
*choisir entre un afficheur lcd I2C de type Digole (PICF182) ou de type LiquidCrystal (PCF8574)*
- `#define LED_PIN 13`

## Fonctions

- `void lectureClavier ()`  
*—lecture clavier—*
- `void temporisationAffichage (const int boucleTemps)`  
*— temporisation pour l'affichage : affichage du menu lorsque temps est > boucleTemps—*
- `void ecritureDateTime ()`  
*—routine lecture et ecriture date and time—*
- `void displayDate ()`  
*—routine display Date—*
- `void displayTime ()`  
*—routine display Time—*
- `void openTime ()`  
*—routine door open time—*
- `void closeTime ()`  
*—routine door close time—*
- `void affiPulsePlusCptRoue ()`  
*—affichage pulse et comptage roue codeuse—*
- `void eclaireAfficheur ()`  
*—retro eclaire de l'afficheur—*
- `void affiTensionBatCdes ()`  
*—affichage tension batterie commandes*
- `void affiTensionBatServo ()`  
*—affichage tension batterie servo-moteur*
- `void affiChoixOuvFerm ()`  
*—affichage du choix de l'ouverture et la fermeture—*
- `void choixOuvFerm ()`  
*—routine choix ouverture fermeture—*
- `void reglageHeureFermeture ()`  
*—reglage de l'heure de fermeture—*
- `void reglageHeureOuverture ()`  
*—routine de reglage de l'heure d'ouverture—*
- `void reglageDate ()`  
*—routine reglage jour semaine, jour, mois, annee—*

```

— void affiLumMatin ()
    ----affichage de la lumiere du matin-----
— void choixLumMatin ()
    ----reglage du choix de la lumiere du matin-----
— void affiLumSoir ()
    ----affichage de la lumiere du soir-----
— void choixLumSoir ()
    ----reglage du choix de la lumiere du soir-----
— void reglageTime ()
    ----routine reglage heure , minute , seconde----
— void affiFinDeCourseFermeture ()
    ----affichage fin de course Fermeture----
— void affiFinDeCourseOuverture ()
    ----affichage fin de course Ouverture-----
— void regFinDeCourseFermeture ()
    ----reglage fin de course Fermeture----
— void regFinDeCourseOuverture ()
    ----reglage fin de course ouverture-----
— void testServo ()
    reglage du servo plus test de la roue codeuse et du servo, à l'aide de la console
— void read_temp (const boolean typeTemperature)
    ----routine lecture température sur ds3231 rtc type celsius=true , fahrenheit=false-----
— void ouverturePorte ()
    ----sequence ouverture de la porte-----
— void fermeturePorte ()
    ----sequence fermeture de la porte----
— void myInterruptINT0 ()
    ----routine interruption D2 INT0-----
— void myInterruptINT1 ()
— void routineInterruptionBp ()
    ----routine interruption Bp----
— void routineInterruptionAlarme2 ()
    ----routine alarme 2----
— void routineInterruptionAlarme1 ()
    ----routine alarme 1----
— void routineTestOuvertureBoitier ()
    ----test ouverture boitier----
— void routineTestFermetureBoitier ()
    ----test fermeture boitier----
— void lumiere ()
    ----routine lecture et affichage de la lumière----
— void ouvFermLum ()
    ----ouverture/fermeture par test de la lumière----
— void deroulementMenu (byte increment)
    ----routine affichage menus-----
— ISR (WDT_vect)
    ----Watchdog Interrupt Service est exécuté lors d'un timeout du WDT----
— void enterSleep (void)
    ----entree dans le mode sleep----
— void routineGestionWatchdog ()
    ----routine de gestion du watchdog----
— void affichageDemarrage (byte colonne)
    ----routine affichage au demarrage----
— void setup ()
— void loop ()

```

## Variables

```

— const boolean debug = false
— const byte bouclesWatchdog = 32
— unsigned int memoireLibre = 0
— volatile int f_wdt = 1
— byte tempsWatchdog = bouclesWatchdog
— boolean reglage = false
— PowerTools tools (debug)
— const boolean emissionRadio = true
— const byte pinEmRadio = 10
— const int vitesseTransmission = 600
— const byte pinSwitchEmissionRadio = A2

```

```

— Radio radio (pinEmRadio, pinSwitchEmissionRadio, vitesseTransmission, VW_MAX_MESSAGE_L↵
  EN, emissionRadio, debug)
— const bool testServoMoteur = false
— const byte servoCde = 8
— const byte servoPin = 4
— const byte securiteHaute = 12
— const int pulseStop = 1500
— bool reduit = false
— ServoMoteur monServo (servoCde, servoPin, securiteHaute, pulseStop, 140, 70, debug)
— const byte accusPinCde = A6
— const byte accusPinServo = A7
— const float tensionMiniAccus = 4.8
— const float rapportConversion = 7.5
— boolean batterieFaible = false
— Accus accusCde (accusPinCde, tensionMiniAccus, rapportConversion, debug)
— Accus accusServo (accusPinServo)
— const byte roueCodeuse = 7
— const unsigned int compteRoueCodeuse = 150
— const unsigned int finDeCourseFermeture = 250
— const unsigned int finDeCourseOuverture = 150
— Codeur codOpt (roueCodeuse, finDeCourseFermeture, finDeCourseOuverture, compteRoue↵
  Codeuse, debug)
— const byte lumierePin = A0
— const float conversion = 5
— const byte heureFenetreSoir = 17
— const byte boucleLumiere = 2
— const unsigned int lumMatin = 300
— const unsigned int lumSoir = 900
— Lumiere lum (lumierePin, lumMatin, lumSoir, heureFenetreSoir, conversion, boucleLumiere, debug)
— volatile boolean interruptBp = false
— volatile boolean interruptRTC = false
— volatile boolean interruptOuvBoi = false
— const byte menuDate = 1
— const byte menuHeure = 2
— const byte menuOuverture = 3
— const byte menuFermeture = 4
— const byte menuTemperature = 5
— const byte menuLumiere = 6
— const byte menuLumiereMatin = 7
— const byte menuLumiereSoir = 8
— const byte menuChoix = 9
— const byte menuFinDeCourseFermeture = 10
— const byte menuFinDeCourseOuverture = 11
— const byte menuTensionBatCdes = 12
— const byte menuTensionBatServo = 13
— const byte menuManuel = 14
— const byte colonnes = 16
— const byte oldkey = -1
— const byte sensorClavier = A1
— const byte pinBp = 9
— const byte pinBoitier = 6
— const int debounce = 200
— byte incrementation = 0
— boolean relache = false
— byte touche = -1
— boolean boitierOuvert = true
— Clavier clav (menuManuel, pinBp, pinBoitier, debounce, debug)
— const int boucleTemps = 100
— bool LcdCursor = true
— int temps = 0
— LcdDigoleI2C mydisp & Wire
  I2C :Arduino UNO : SDA (data line) is on analog input pin 4, and SCL (clock line) is on analog input pin 5 on UNO
  and Duemilanove.
— const byte rtclINT = 5
— const byte adresseBoitier24C32 = 0x57
— const byte jourSemaine = 1
— const byte jour = 2
— const byte mois = 3
— const byte annee = 4
— const byte heure = 5
— const byte minutes = 6
— const byte secondes = 7
— const byte alarm_1 = 1
— const byte alarm_2 = 2
— const bool typeTemperature = true
— tmElements_t tm

```

- **HorlogeDS3232 rtc (adresseBoitier24C32, rtcINT, debug)**
- `const char listeDayWeek[] PROGMEM = "DimLunMarMerJeuVenSam"`

## 6.20.1 Documentation des macros

### 6.20.1.1 #define LCD\_DIGOLE

choisir entre un afficheur lcd I2C de type Digole (PICF182) ou de type LiquidCrystal (PCF8574)

porte-poulailler : affichage + clavier + rtc 01 2017 classes radio, lcd (digole et liquidcrystal), horloge, bouton. 29 12 2016 classe **Codeur** (p. ??) (optique). 28 12 2016 classe **Lumiere** (p. ??) - ok. 26 12 2016 classe **Accus** (p. ??) - ok. 21 12 2016 ajout de la classe servo - ok. 01 12 2016first commit sur github. 20 09 2016 classe clavier/

19 septembre 2016 :

- suppression du blocage au demarrage (si = 200). 18 septembre 2016 : En cas de probleme avec le sens de rotation du servo :
- variable globale : sens.
- arret fin de course ( contact) quelque soit le sens.
- diminution de la vitesse reduite dans la descente de la trappe (inertie). 17 septembre 2016 :

problème avec servotimer2 : Le pulse en ms ne fonctionne pas donc le servo est tjs dans le même sens (hs). Les valeurs du pulse étaient mauvaise. Il faut : 90° : 1500ms. 0° : 1000ms. 180° : 2000ms. 10 septembre 2016 :

- affichage compteur roue codeuse envoi radio.
- delay d'attente de la remontée du servo augmentée : 300. Important : -with ServoTimer2 et VirtualWire. -il faut commenter typedef dans la bibliothèque servotimer2.

Définition à la ligne 48 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

### 6.20.1.2 #define LED\_PIN 13

led broche 13

Définition à la ligne 80 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

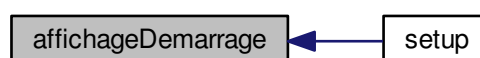
## 6.20.2 Documentation des fonctions

### 6.20.2.1 void affichageDemarrage ( byte colonne )

---routine affichage au demarrage---

Définition à la ligne 967 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe des appelants de cette fonction :

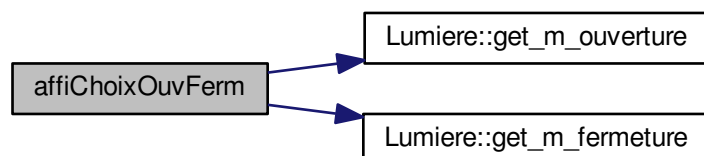


### 6.20.2.2 void affiChoixOuvFerm ( )

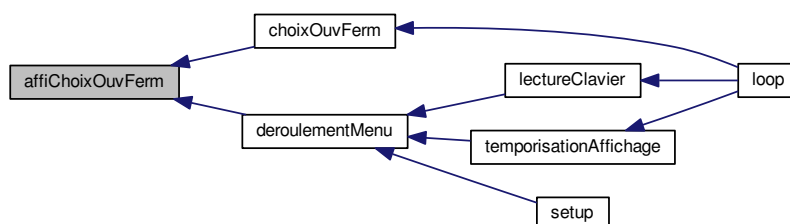
-----affichage du choix de l'ouverture et la fermeture-----

Définition à la ligne 338 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

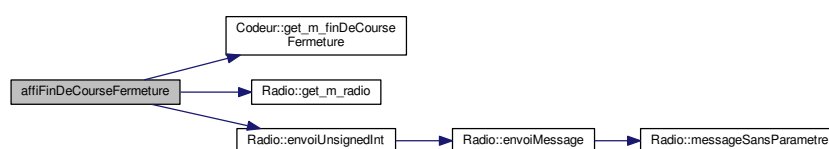


### 6.20.2.3 void affiFinDeCourseFermeture ( )

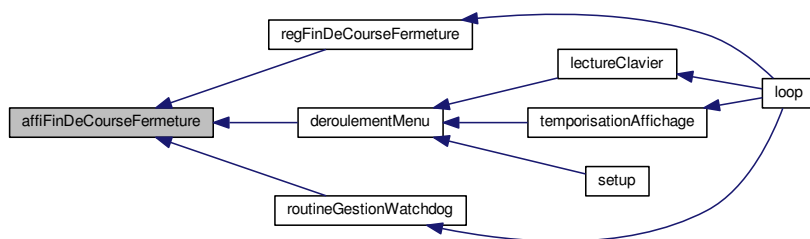
-----affichage fin de course Fermeture-----

Définition à la ligne 534 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

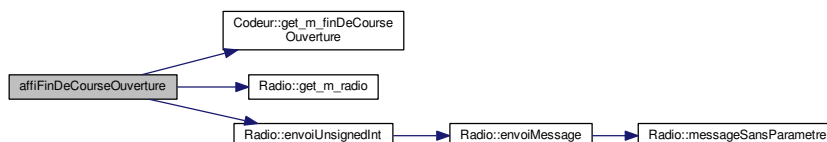


#### 6.20.2.4 void affiFinDeCourseOuverture ( )

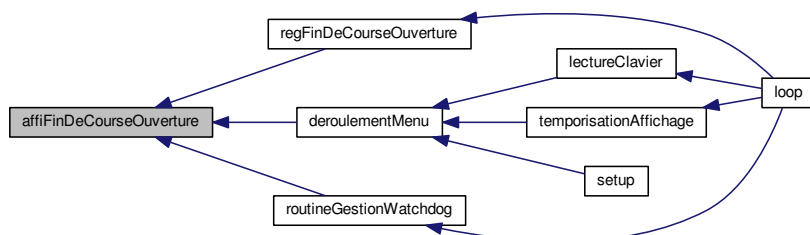
-----affichage fin de course Ouverture-----

Définition à la ligne 545 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 6.20.2.5 void affiLumMatin ( )

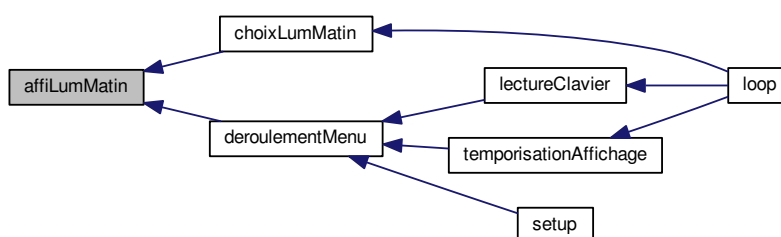
-----affichage de la lumiere du matin-----

Définition à la ligne 448 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 6.20.2.6 void affiLumSoir ( )

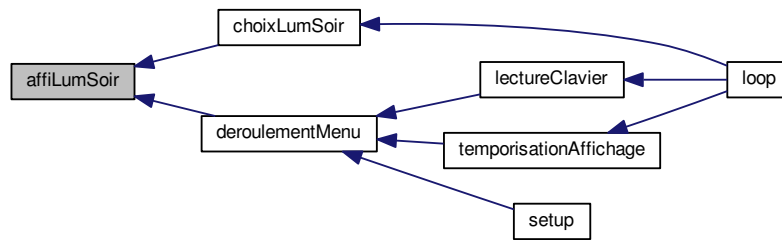
-----affichage de la lumiere du soir-----

Définition à la ligne 478 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

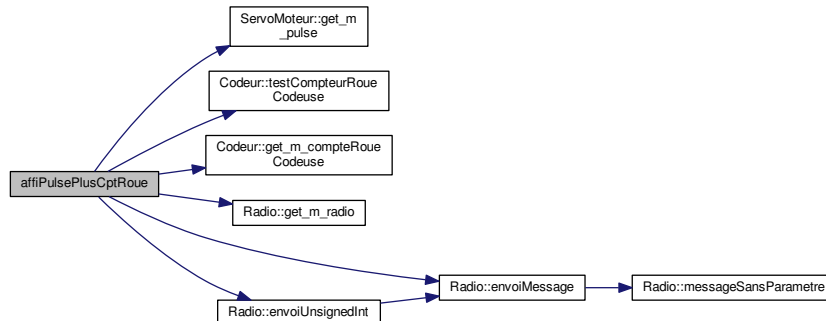


#### 6.20.2.7 void affiPulsePlusCptRoue ( )

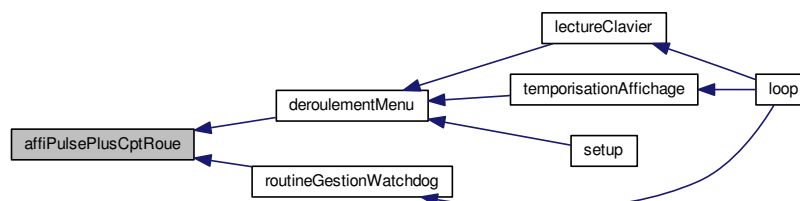
-----affichage pulse et comptage roue codeuse-----

Définition à la ligne 270 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



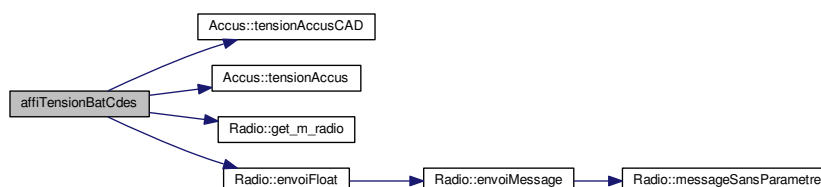


#### 6.20.2.8 void affiTensionBatCdes ( )

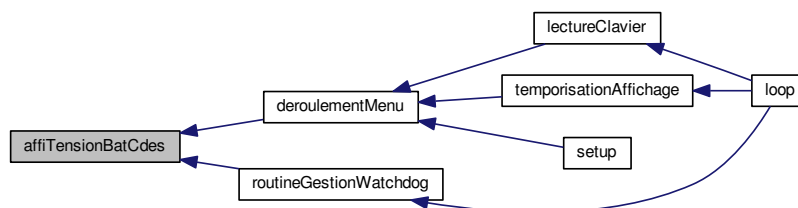
-----affichage tension batterie commandes

Définition à la ligne 308 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

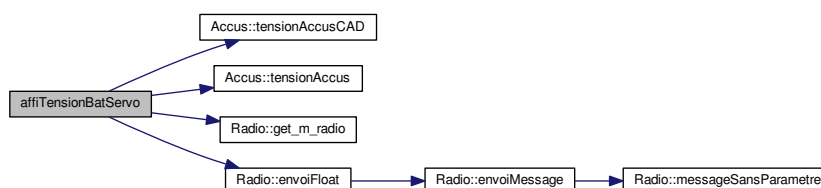


#### 6.20.2.9 void affiTensionBatServo ( )

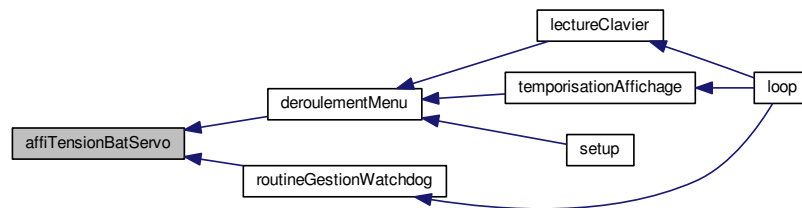
-----affichage tension batterie servo-moteur

Définition à la ligne 321 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

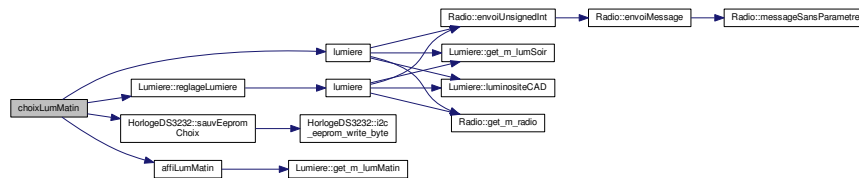


#### 6.20.2.10 void choixLumMatin ( )

-----reglage du choix de la lumiere du matin-----

Définition à la ligne 457 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

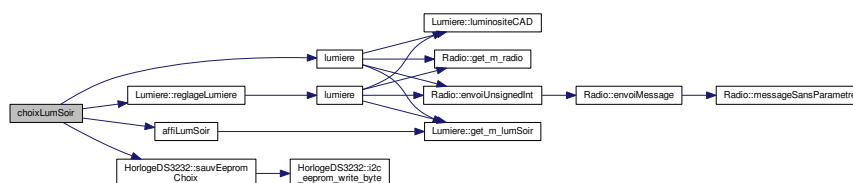


#### 6.20.2.11 void choixLumSoir ( )

-----reglage du choix de la lumiere du soir-----

Définition à la ligne 487 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

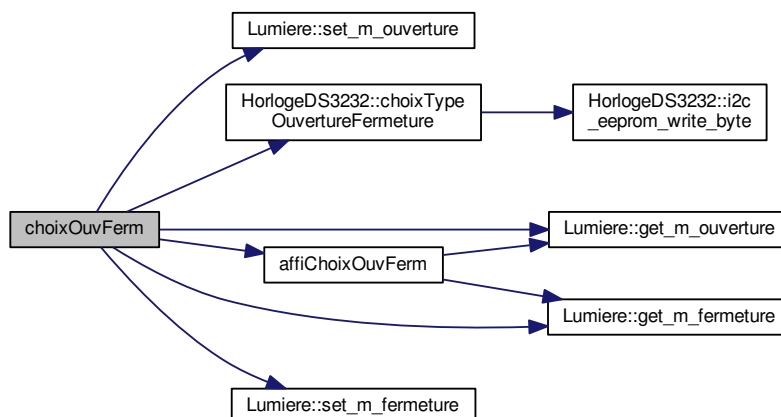


#### 6.20.2.12 void choixOuvFerm ( )

---routine choix ouverture fermeture---

Définition à la ligne 346 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

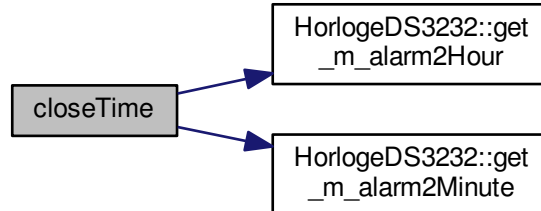


#### 6.20.2.13 void closeTime ( )

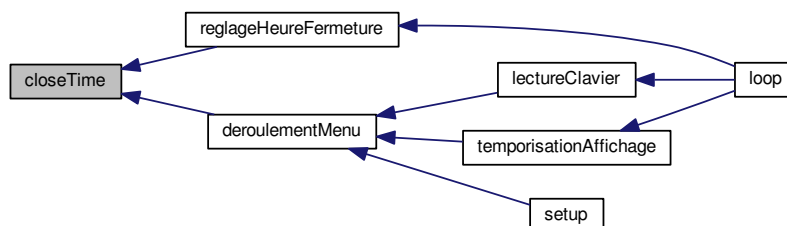
---routine door close time---

Définition à la ligne 262 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

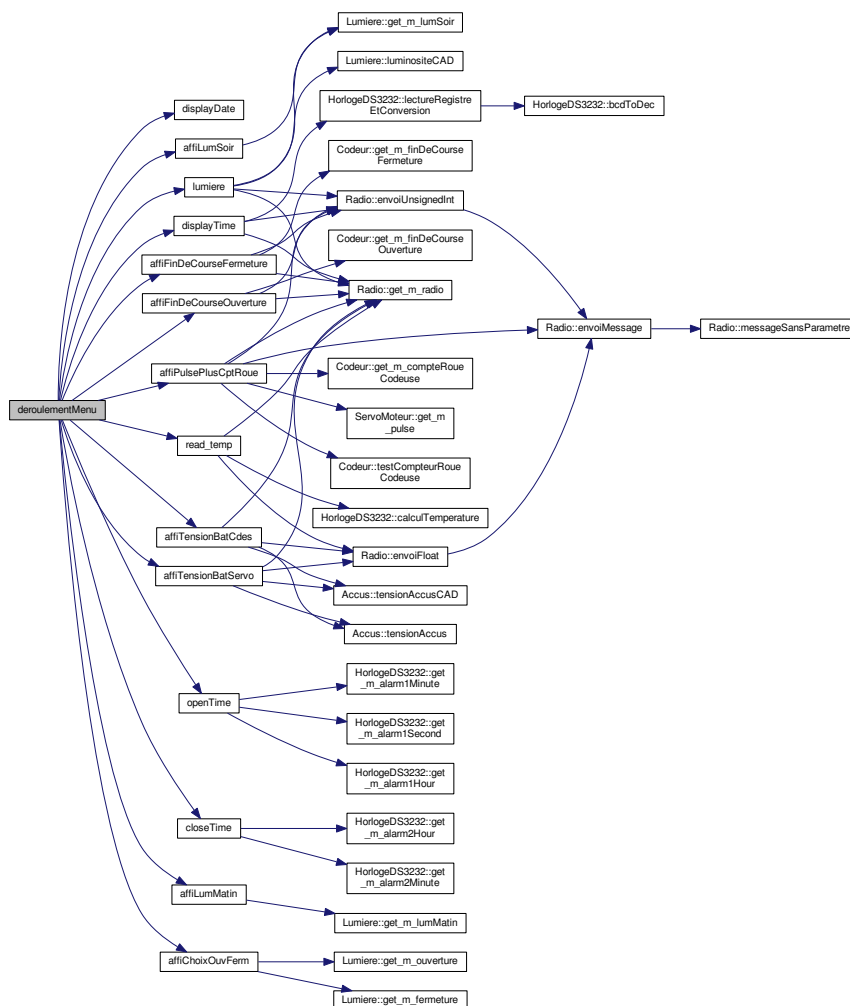


### 6.20.2.14 void deroulementMenu ( byte *increment* )

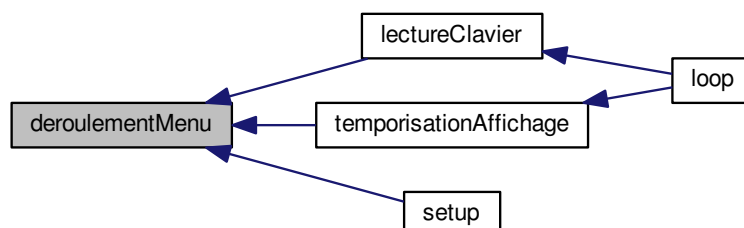
---routine affichage menus---

Définition à la ligne 855 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

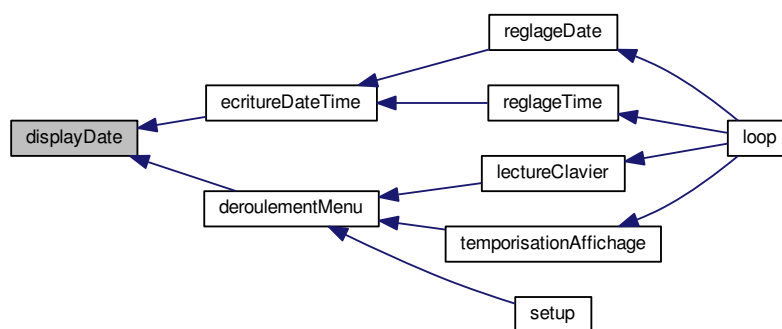


#### 6.20.2.15 void displayDate ( )

---routine display Date---

Définition à la ligne 227 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe des appelants de cette fonction :

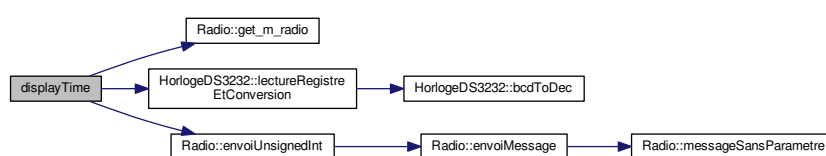


#### 6.20.2.16 void displayTime ( )

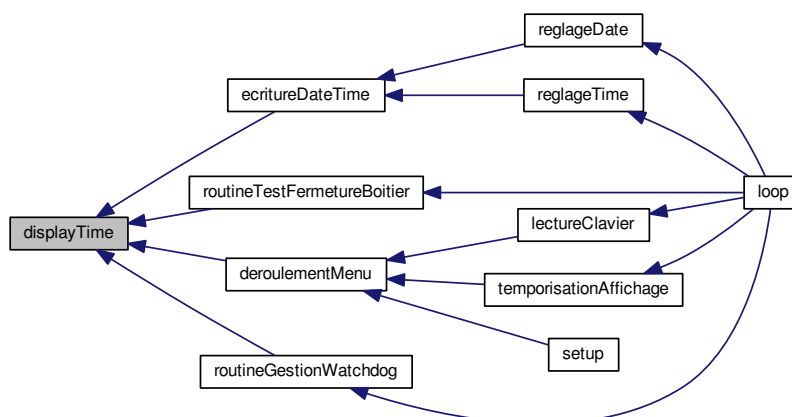
---routine display Time---

Définition à la ligne 241 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 6.20.2.17 void eclaireageAfficheur ( )

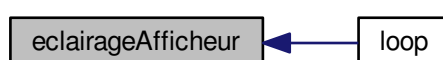
---retro eclairage de l'afficheur---

Définition à la ligne 298 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

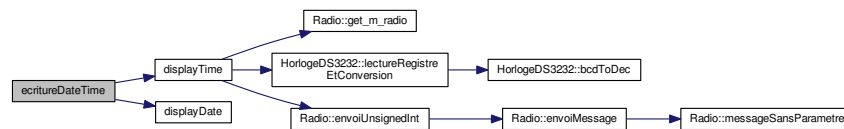


### 6.20.2.18 void ecritureDateTime ( )

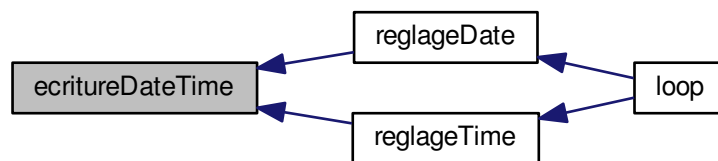
—routine lecture et ecriture date and time—

Définition à la ligne 215 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

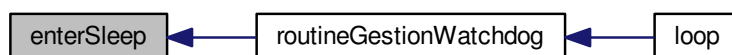


### 6.20.2.19 void enterSleep ( void )

—entree dans le mode sleep—

Définition à la ligne 919 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe des appelants de cette fonction :



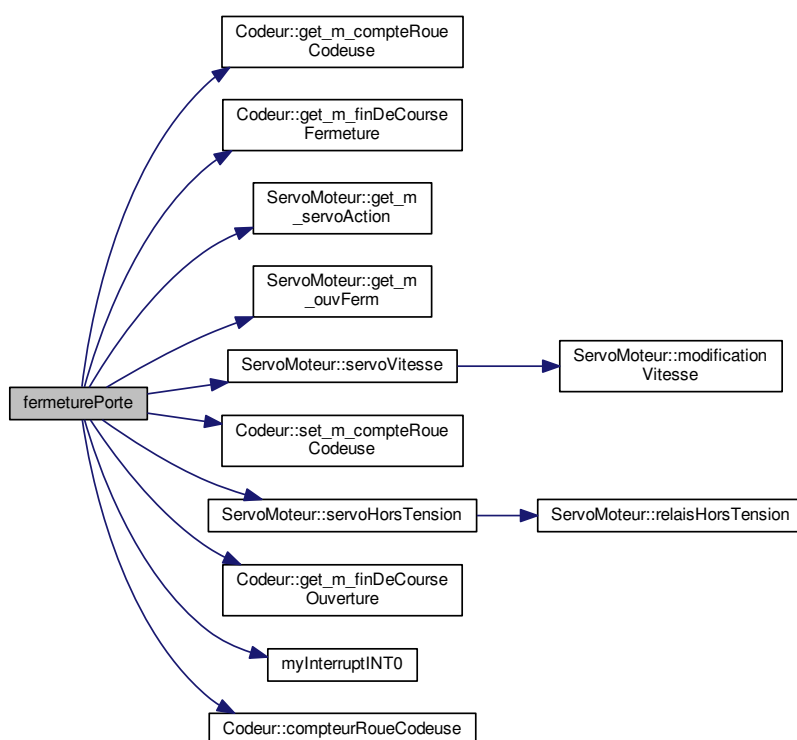


#### 6.20.2.20 void fermeturePorte ( )

—sequence fermeture de la porte—

Définition à la ligne 710 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 6.20.2.21 ISR ( WDT\_vect )

—Watchdog Interrupt Service est exécuté lors d'un timeout du WDT—

Définition à la ligne 912 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

### 6.20.2.22 void lectureClavier ( )

—lecture clavier—

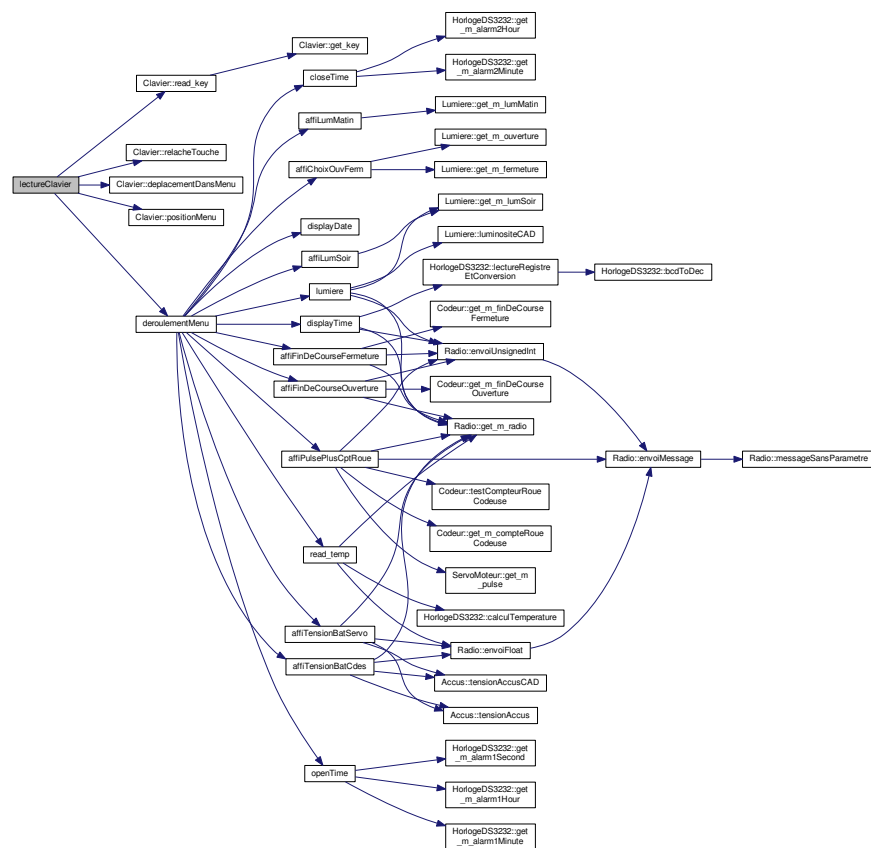
read key sensor = A1

position du menu pour l'affichage

affichage du menu

Définition à la ligne 193 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



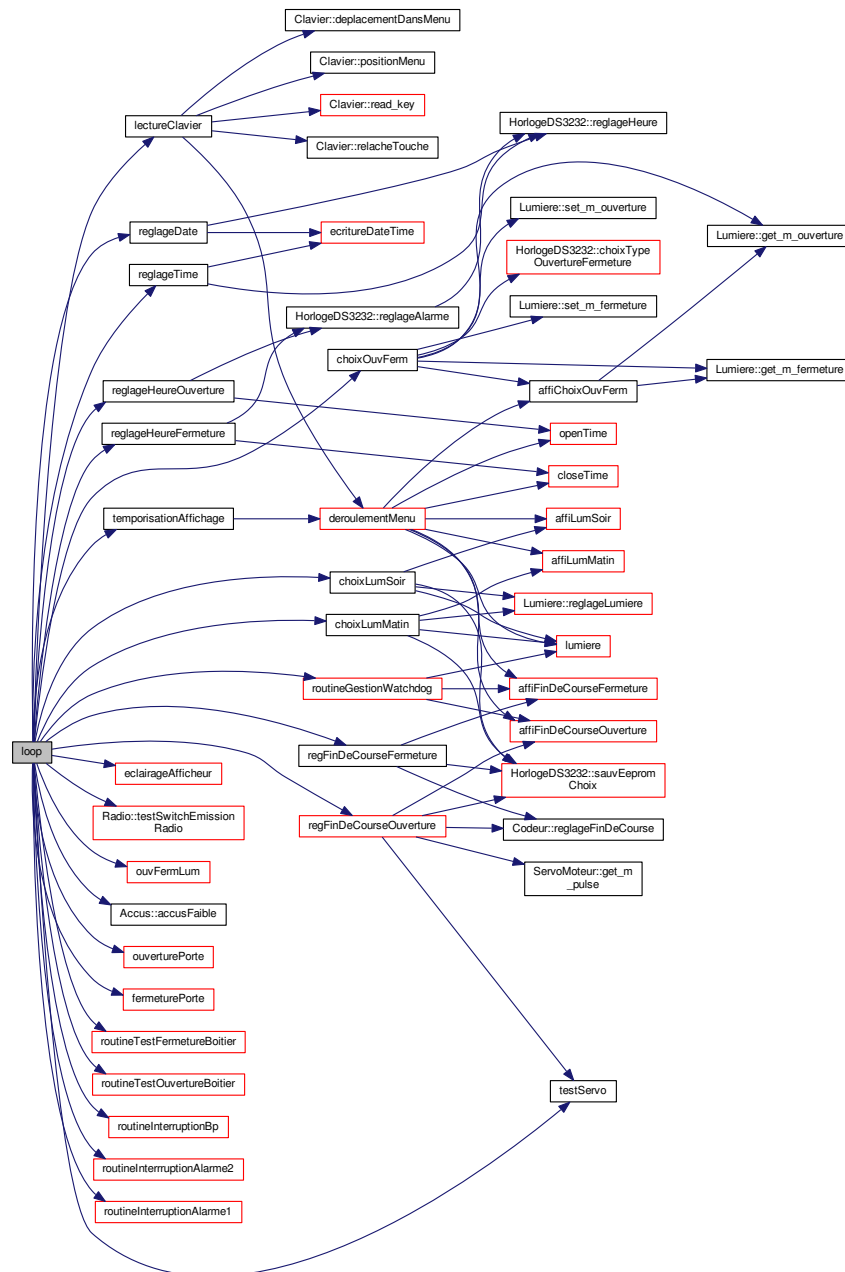
Voici le graphe des appelants de cette fonction :



### 6.20.2.23 void loop ( )

Définition à la ligne 1050 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :

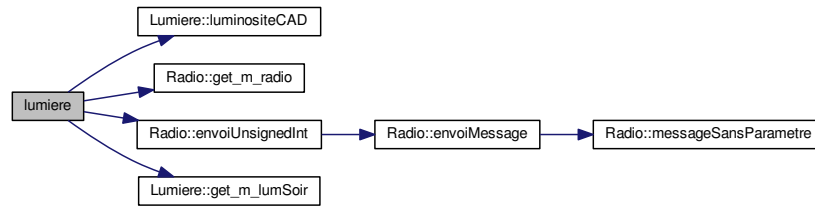


### 6.20.2.24 void lumiere ( )

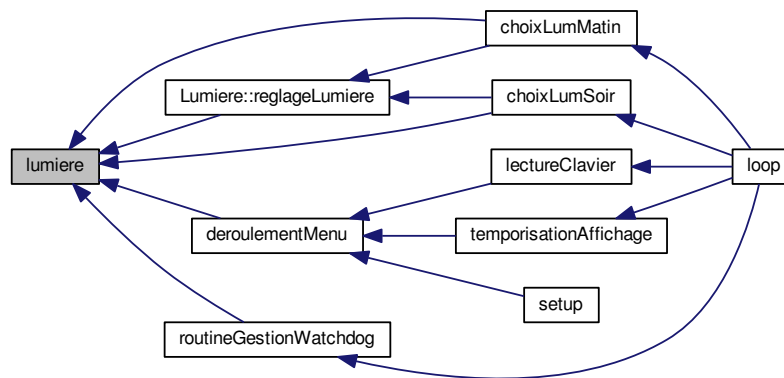
---routine lecture et affichage de la lumière---

Définition à la ligne 818 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



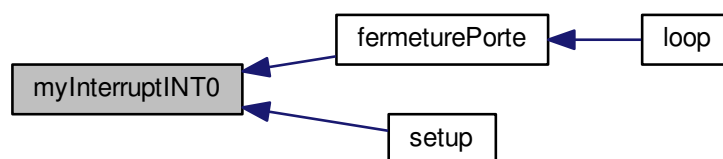
#### 6.20.2.25 void myInterruptINT0 ( )

---routine interruption D2 INT0---

interruptions -routine interruption D2 INT0 -routine interruption D3 INT1 -routine interruption Bp -routine alarme 2  
-routine alarme 1 -routine interruption boitier ouvert -test fermeture boitier

Définition à la ligne 734 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

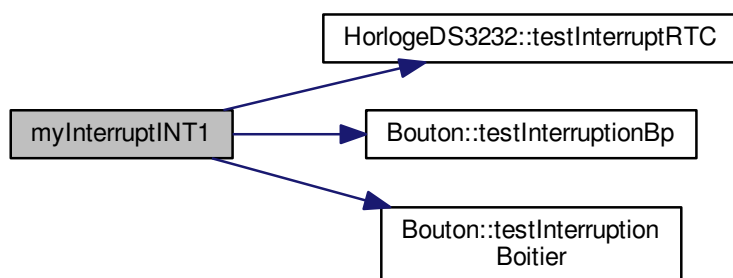
Voici le graphe des appelants de cette fonction :



#### 6.20.2.26 void myInterruptINT1 ( )

Définition à la ligne 739 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

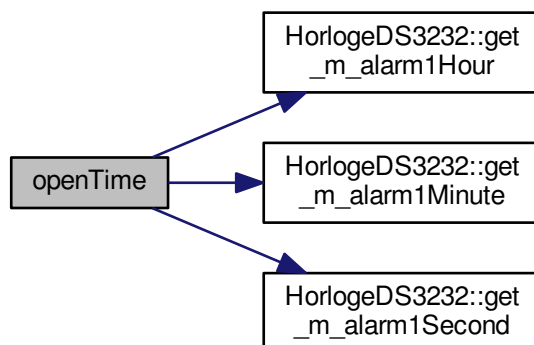


#### 6.20.2.27 void openTime ( )

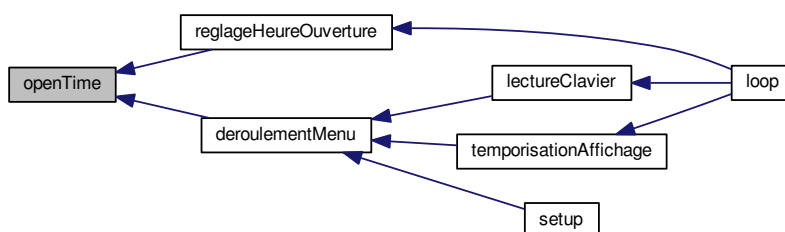
—routine door open time—

Définition à la ligne 254 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

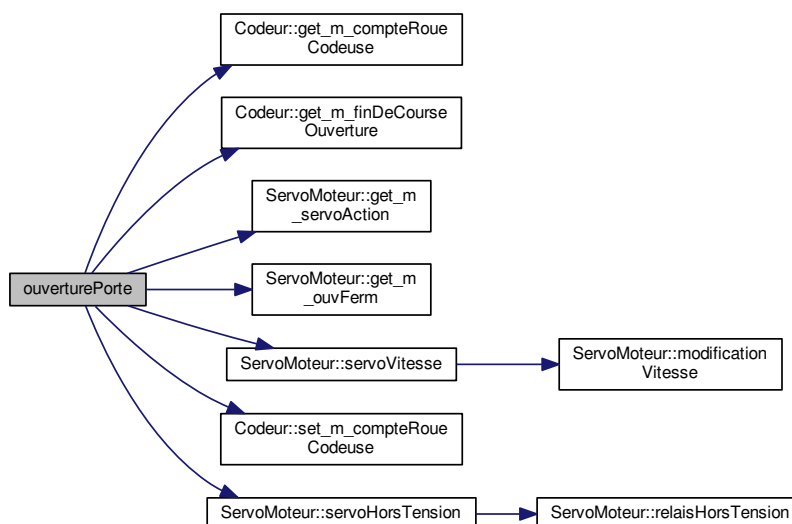


#### 6.20.2.28 void ouverturePorte ( )

-----sequence ouverture de la porte-----

Définition à la ligne 695 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

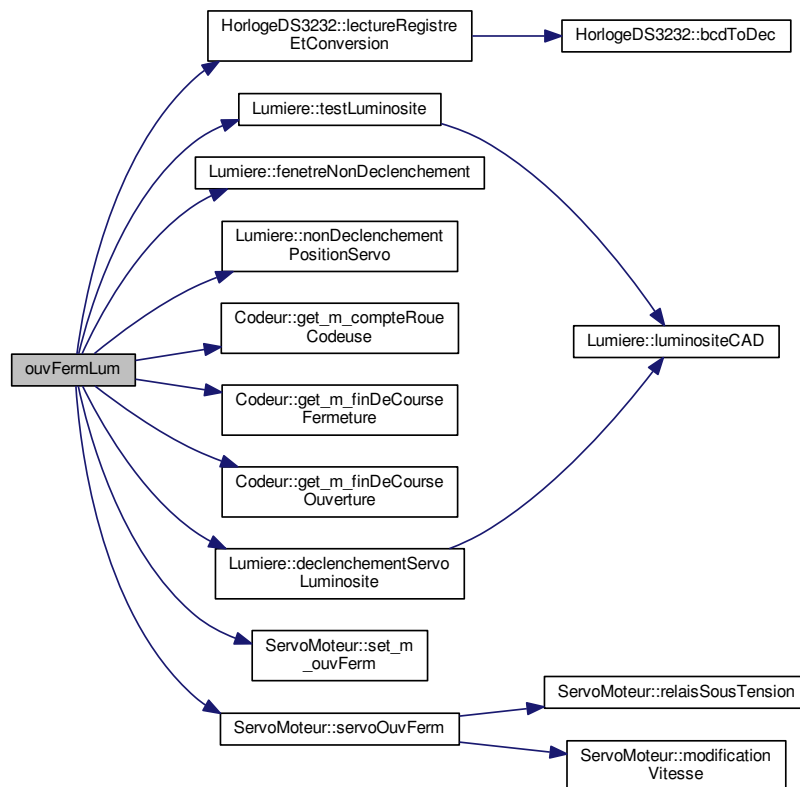


#### 6.20.2.29 void ouvFermLum ( )

---ouverture/fermeture par test de la lumière---

Définition à la ligne 831 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



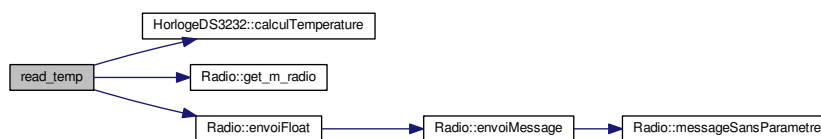
#### 6.20.2.30 void read\_temp ( const boolean typeTemperature )

-----routine lecture température sur ds3231 rtc type celsius=true ,fahrenheit=false-----

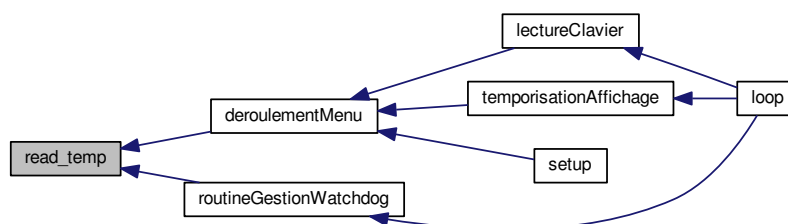
Définition à la ligne 676 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.



Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

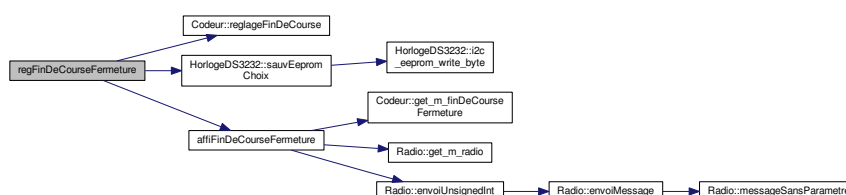


#### 6.20.2.31 void regFinDeCourseFermeture ( )

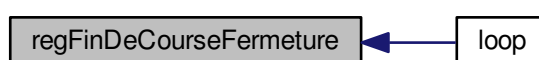
-----reglage fin de course Fermeture-----

Définition à la ligne 556 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

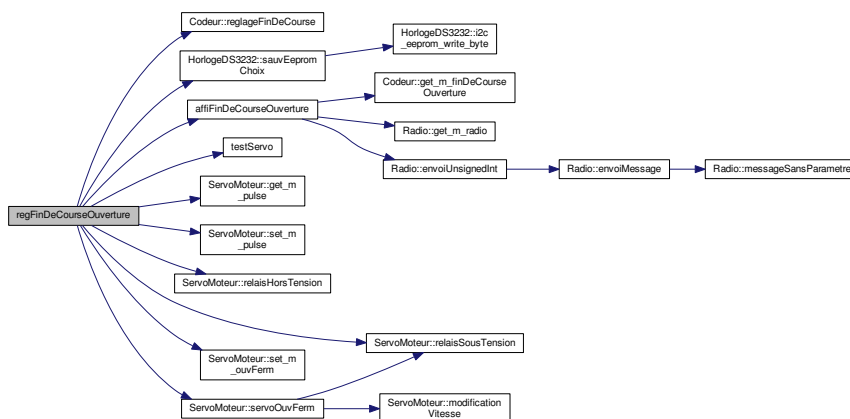


### 6.20.2.32 void regFinDeCourseOuverture ( )

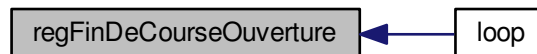
---regalge fin de course ouverture---

Définition à la ligne 576 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

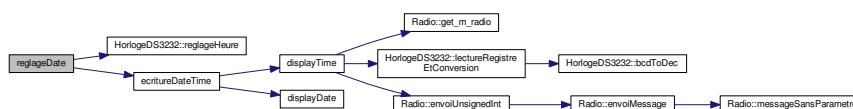


### 6.20.2.33 void reglageDate ( )

---routine reglage jour semaine, jour, mois, annee---

Définition à la ligne 418 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

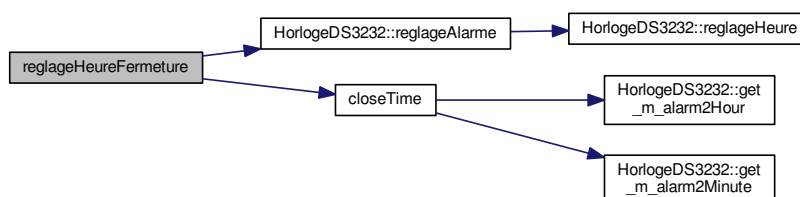


#### 6.20.2.34 void reglageHeureFermeture ( )

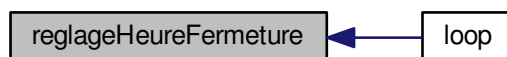
----reglage de l'heure de fermeture----

Définition à la ligne 368 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

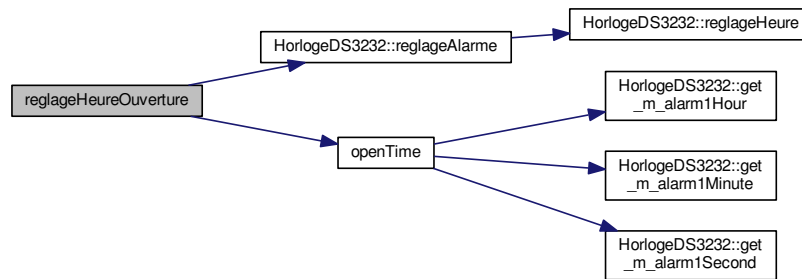


#### 6.20.2.35 void reglageHeureOuverture ( )

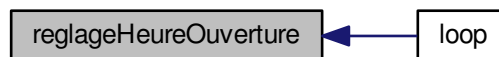
----routine de reglage de l'heure d'ouverture----

Définition à la ligne 391 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

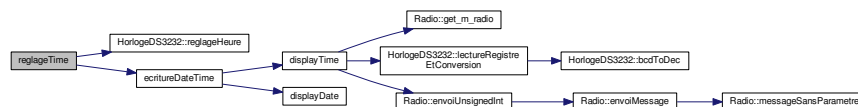


#### 6.20.2.36 void reglageTime ( )

---routine reglage heure , minute , seconde---

Définition à la ligne 508 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

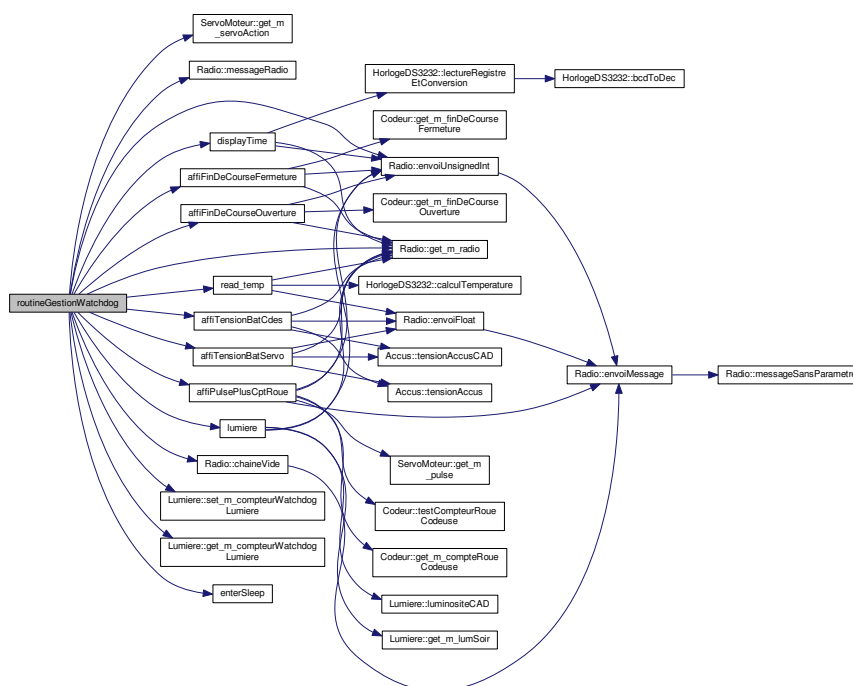


### 6.20.2.37 void routineGestionWatchdog ( )

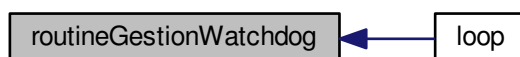
---routine de gestion du watchdog---

Définition à la ligne 928 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

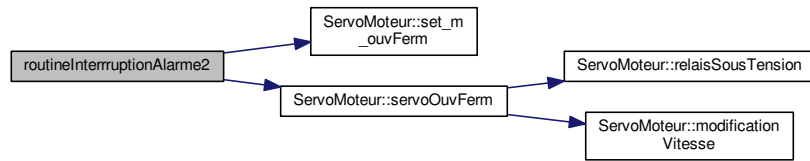


### 6.20.2.38 void routineInterruptionAlarme2 ( )

---routine alarme 2---

Définition à la ligne 758 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

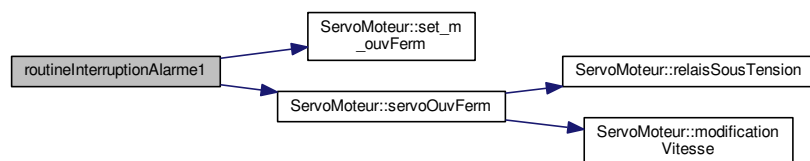


#### 6.20.2.39 void routineInterruptionAlarme1 ( )

---routine alarme 1---

Définition à la ligne 769 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

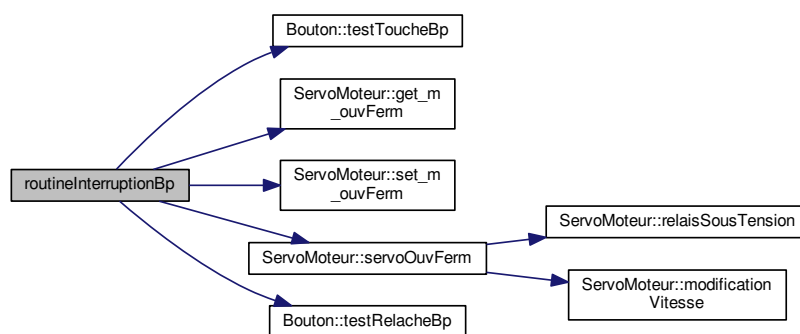


#### 6.20.2.40 void routineInterruptionBp ( )

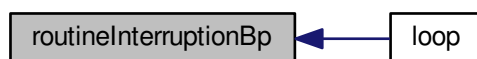
---routine interruption Bp---

Définition à la ligne 746 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

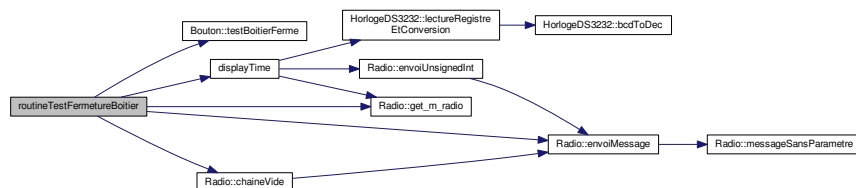


#### 6.20.2.41 void routineTestFermetureBoitier ( )

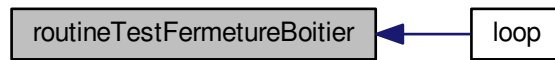
---test fermeture boitier---

Définition à la ligne 796 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

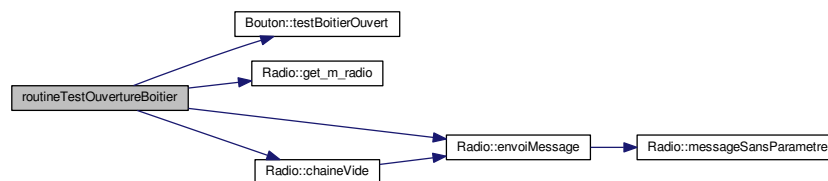


#### 6.20.2.42 void routineTestOuvertureBoitier ( )

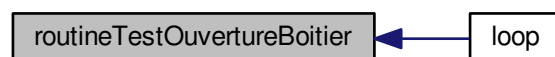
---test ouverture boitier---

Définition à la ligne 780 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :

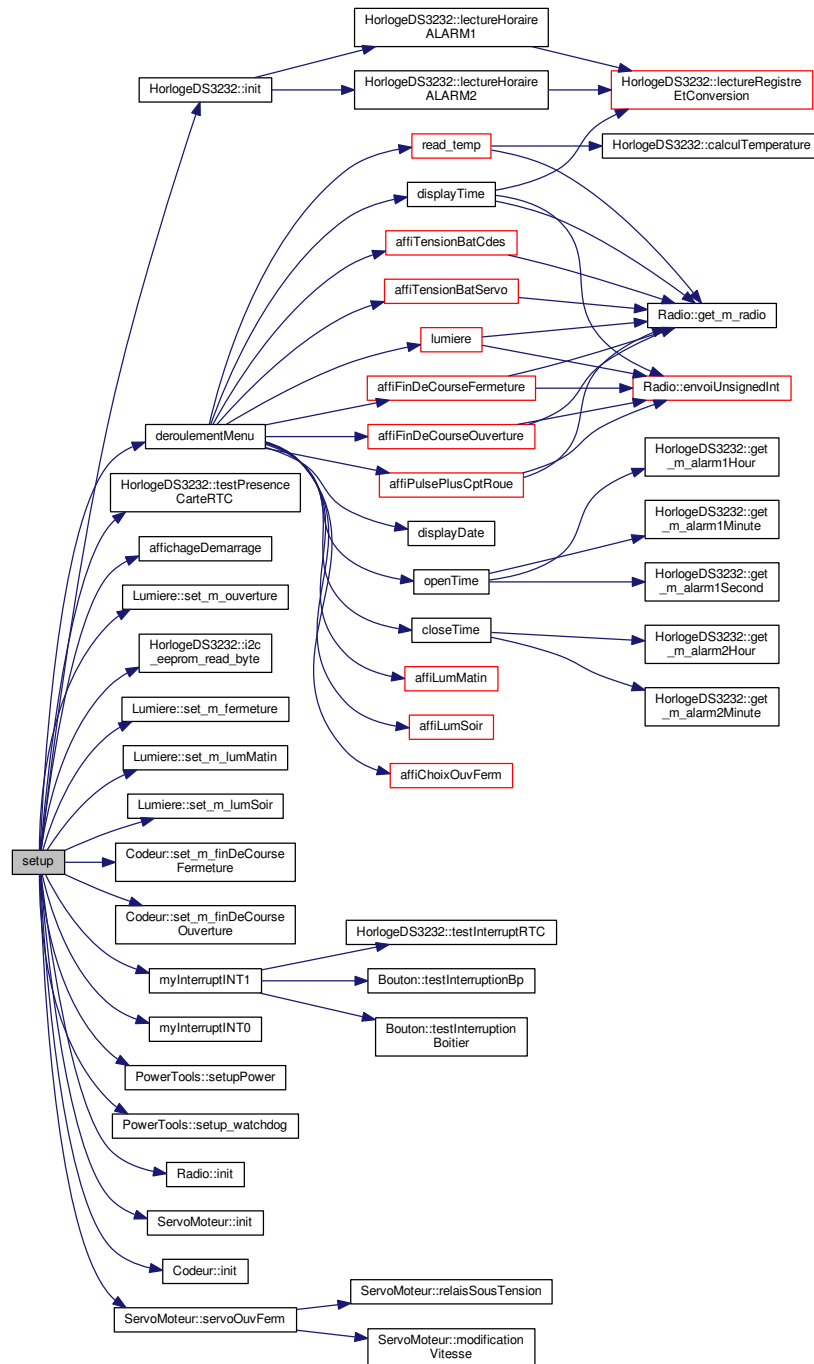


#### 6.20.2.43 void setup ( )

Définition à la ligne 981 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.



Voici le graphe d'appel pour cette fonction :

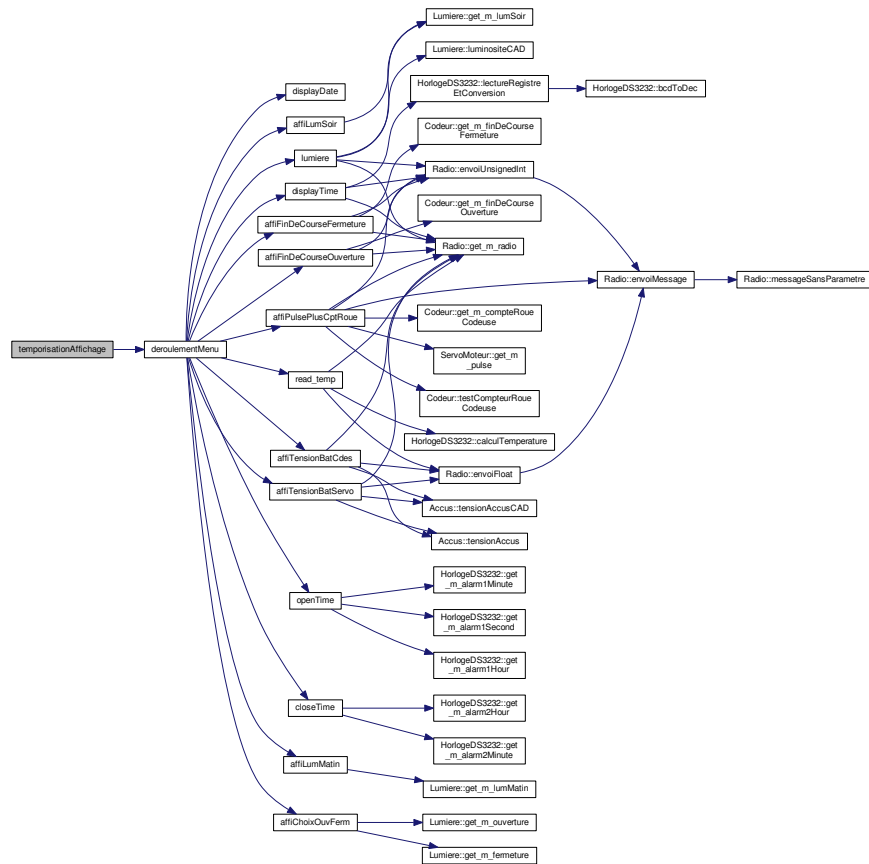


#### 6.20.2.44 void temporisationAffichage ( const int boucleTemps )

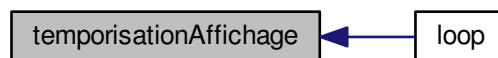
— temporisation pour l'affichage : affichage du menu lorsque temps est > boucleTemps—

Définition à la ligne 205 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe d'appel pour cette fonction :



Voici le graphe des appelants de cette fonction :



#### 6.20.2.45 void testServo ( )

reglage du servo plus test de la roue codeuse et du servo, à l'aide de la console

Test du servo avec la console arduino

a = -1 q = -1

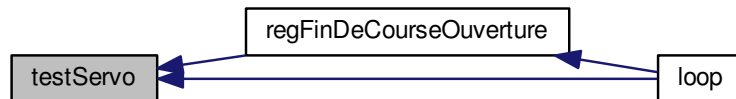
z = +10 s = -10

e = arrêt du servo (relais) d = marche du servo (relais)

r = montée f = descente

Définition à la ligne 611 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

Voici le graphe des appelants de cette fonction :



### 6.20.3 Documentation des variables

6.20.3.1 **Accus** accusCde(accusPinCde, tensionMiniAccus, rapportConversion, debug)

6.20.3.2 **const byte** accusPinCde = A6

**Accus** (p. ??)

Définition à la ligne 95 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.3 **const byte** accusPinServo = A7

Définition à la ligne 96 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.4 **Accus** accusServo(accusPinServo)

6.20.3.5 **const byte** adresseBoitier24C32 = 0x57

Définition à la ligne 173 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.6 **const byte** alarm\_1 = 1

Définition à la ligne 175 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.7 **const byte** alarm\_2 = 2

Définition à la ligne 176 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.8 `const byte annee = 4`

Définition à la ligne 174 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

6.20.3.9 `boolean batterieFaible = false`

Définition à la ligne 99 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

6.20.3.10 `boolean boitierOuvert = true`

Définition à la ligne 152 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

6.20.3.11 `const byte boucleLumiere = 2`

Définition à la ligne 116 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

6.20.3.12 `const byte bouclesWatchdog = 32`

Définition à la ligne 63 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

6.20.3.13 `const int boucleTemps = 100`

LCD DigoleSerialI2C

Définition à la ligne 156 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

6.20.3.14 `Clavier clav(menuManuel, pinBp, pinBoitier, debounce, debug)`

6.20.3.15 `Codeur codOpt(roueCodeuse, finDeCourseFermeture, finDeCourseOuverture, compteRoueCodeuse, debug)`

6.20.3.16 `const byte colonnes = 16`

Définition à la ligne 143 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

6.20.3.17 `const unsigned int compteRoueCodeuse = 150`

Définition à la ligne 106 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

6.20.3.18 `const float conversion = 5`

Définition à la ligne 114 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

6.20.3.19 `const int debounce = 200`

Définition à la ligne 148 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.20 `const boolean debug = false`

——Bibliothèque Flash pour mise en mémoire flash de l'arduino F()—— power and tools watchdog - Optimisation de la consommation

Définition à la ligne 61 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.21 `const boolean emissionRadio = true`

radio 433MHz

Définition à la ligne 73 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.22 `volatile int f_wdt = 1`

Définition à la ligne 66 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.23 `const unsigned int finDeCourseFermeture = 250`

Définition à la ligne 107 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.24 `const unsigned int finDeCourseOuverture = 150`

Définition à la ligne 108 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.25 `const byte heure = 5`

Définition à la ligne 174 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.26 `const byte heureFenetreSoir = 17`

Définition à la ligne 115 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.27 `byte incrementation = 0`

Définition à la ligne 149 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

**6.20.3.28** volatile boolean interruptBp = false

interruptions

Définition à la ligne 122 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

**6.20.3.29** volatile boolean interruptOuvBoi = false

Définition à la ligne 124 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

**6.20.3.30** volatile boolean interruptRTC = false

Définition à la ligne 123 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

**6.20.3.31** const byte jour = 2

Définition à la ligne 174 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

**6.20.3.32** const byte jourSemaine = 1

Définition à la ligne 174 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

**6.20.3.33** bool LcdCursor = true

Définition à la ligne 157 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

**6.20.3.34** Lumiere lum(lumierePin, lumMatin, lumSoir, heureFenetreSoir, conversion, boucleLumiere, debug)

**6.20.3.35** const byte lumierePin = A0

lumiere

Définition à la ligne 113 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

**6.20.3.36** const unsigned int lumMatin = 300

Définition à la ligne 117 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

**6.20.3.37** const unsigned int lumSoir = 900

Définition à la ligne 118 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.38 unsigned int memoireLibre = 0

Définition à la ligne 65 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.39 const byte menuChoix = 9

Définition à la ligne 137 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.40 const byte menuDate = 1

**Clavier** (p. ??)

Définition à la ligne 129 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.41 const byte menuFermeture = 4

Définition à la ligne 132 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.42 const byte menuFinDeCourseFermeture = 10

Définition à la ligne 138 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.43 const byte menuFinDeCourseOuverture = 11

Définition à la ligne 139 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.44 const byte menuHeure = 2

Définition à la ligne 130 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.45 const byte menuLumiere = 6

Définition à la ligne 134 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.46 const byte menuLumiereMatin = 7

Définition à la ligne 135 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.47 const byte menuLumiereSoir = 8

Définition à la ligne 136 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

**6.20.3.48** `const byte menuManuel = 14`

Définition à la ligne 142 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

**6.20.3.49** `const byte menuOuverture = 3`

Définition à la ligne 131 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

**6.20.3.50** `const byte menuTemperature = 5`

Définition à la ligne 133 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

**6.20.3.51** `const byte menuTensionBatCdes = 12`

Définition à la ligne 140 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

**6.20.3.52** `const byte menuTensionBatServo = 13`

Définition à la ligne 141 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

**6.20.3.53** `const byte minutes = 6`

Définition à la ligne 174 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

**6.20.3.54** `const byte mois = 3`

Définition à la ligne 174 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

**6.20.3.55** `ServoMoteur monServo(servoCde, servoPin, securiteHaute, pulseStop, 140, 70, debug)`

**6.20.3.56** `const byte oldkey = -1`

Définition à la ligne 144 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

**6.20.3.57** `const byte pinBoitier = 6`

Définition à la ligne 147 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.

**6.20.3.58** `const byte pinBp = 9`

Définition à la ligne 146 du fichier `porte_poulailler_affichage_clavier_rtc.ino`.



6.20.3.59 `const byte pinEmRadio = 10`

Définition à la ligne 74 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.60 `const byte pinSwitchEmissionRadio = A2`

Définition à la ligne 76 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.61 `const char affichageBonjour [ ] PROGMEM = "DimLunMarMerJeuVenSam"`

progmem mémoire flash

Définition à la ligne 182 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.62 `const int pulseStop = 1500`

Définition à la ligne 88 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.63 `Radio radio(pinEmRadio, pinSwitchEmissionRadio, vitesseTransmission, VW_MAX_MESSAGE_LEN, emissionRadio, debug)`

6.20.3.64 `const float rapportConversion = 7.5`

Définition à la ligne 98 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.65 `bool reduit = false`

Définition à la ligne 89 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.66 `boolean reglage = false`

Définition à la ligne 68 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.67 `boolean relache = false`

Définition à la ligne 150 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.68 `const byte roueCodeuse = 7`

roue codeuse

Définition à la ligne 105 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.69 **HorlogeDS3232 rtc(adresseBoitier24C32, rtcINT, debug)**

6.20.3.70 **const byte rtcINT = 5**

RTC\_DS3231

Définition à la ligne 172 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.71 **const byte secondes = 7**

Définition à la ligne 174 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.72 **const byte securiteHaute = 12**

Définition à la ligne 87 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.73 **const byte sensorClavier = A1**

Définition à la ligne 145 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.74 **const byte servoCde = 8**

Définition à la ligne 85 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.75 **const byte servoPin = 4**

Définition à la ligne 86 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.76 **int temps = 0**

Définition à la ligne 158 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.77 **byte tempsWatchdog = bouclesWatchdog**

Définition à la ligne 67 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.78 **const float tensionMiniAccus = 4.8**

Définition à la ligne 97 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.79 `const bool testServoMoteur = false`

servo - montée et descente de la porte

Définition à la ligne 84 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.80 `tmElements_t tm`

Définition à la ligne 178 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.81 `PowerTools tools(debug)`

6.20.3.82 `byte touche = -1`

Définition à la ligne 151 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.83 `const bool typeTemperature = true`

Définition à la ligne 177 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.84 `const int vitesseTransmission = 600`

Définition à la ligne 75 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

6.20.3.85 `LcdDigoleI2C mydisp & Wire`

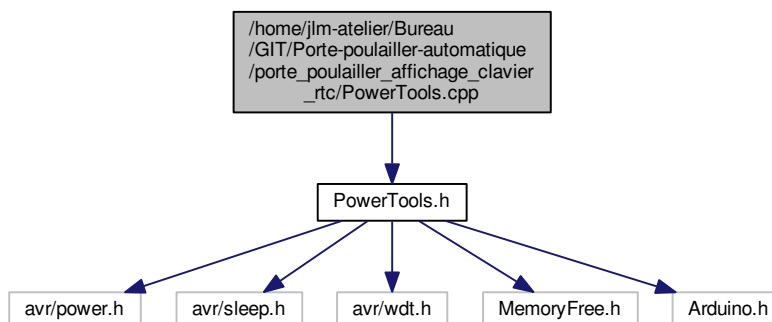
I2C :Arduino UNO : SDA (data line) is on analog input pin 4, and SCL (clock line) is on analog input pin 5 on UNO and Duemilanove.

Définition à la ligne 162 du fichier porte\_poulailler\_affichage\_clavier\_rtc.ino.

## 6.21 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_↵ \_poulailler\_affichage\_clavier\_rtc/PowerTools.cpp

```
#include "PowerTools.h"
```

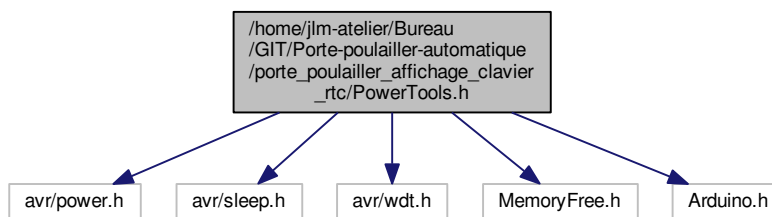
Graphe des dépendances par inclusion de PowerTools.cpp :



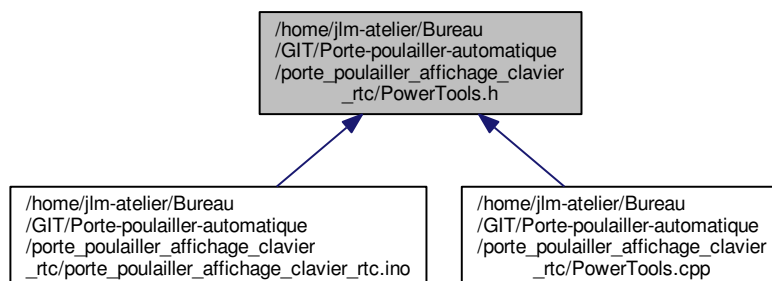
## 6.22 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/PowerTools.h

```
#include <avr/power.h>
#include <avr/sleep.h>
#include <avr/wdt.h>
#include <MemoryFree.h>
#include "Arduino.h"
```

Graphe des dépendances par inclusion de PowerTools.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



### Classes

— class **PowerTools**

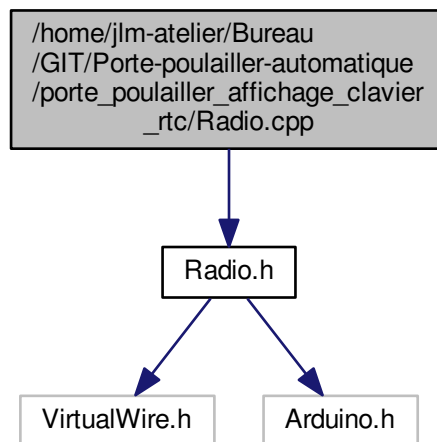
## 6.23 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/Radio.cpp

```
#include "Radio.h"
```

## 6.24 Référence du fichier

/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/Radio.t85

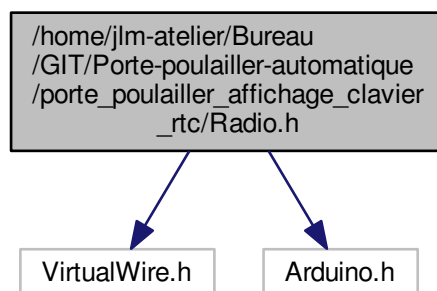
Graphe des dépendances par inclusion de Radio.cpp :



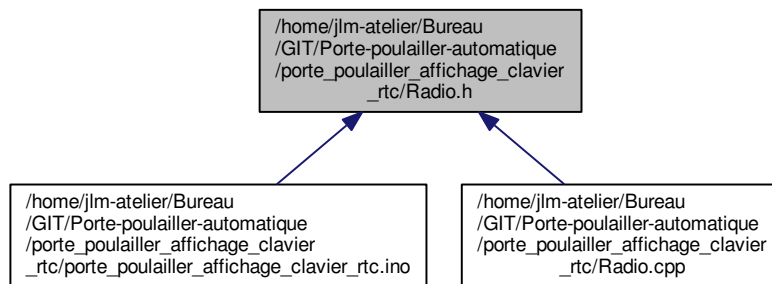
## 6.24 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/Radio.h

```
#include <VirtualWire.h>  
#include "Arduino.h"
```

Graphe des dépendances par inclusion de Radio.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



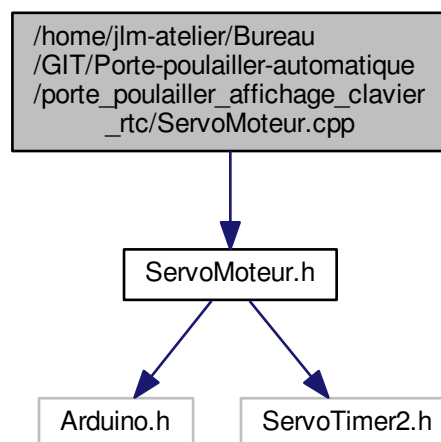
## Classes

— class **Radio**

### 6.25 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_poulailler\_affichage\_clavier\_rtc/ServoMoteur.cpp

```
#include "ServoMoteur.h"
```

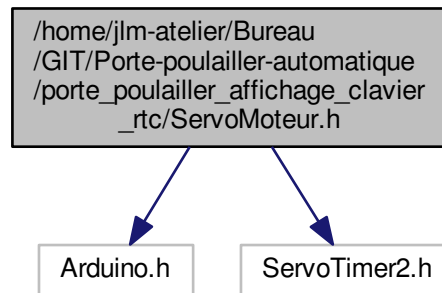
Graphe des dépendances par inclusion de ServoMoteur.cpp :



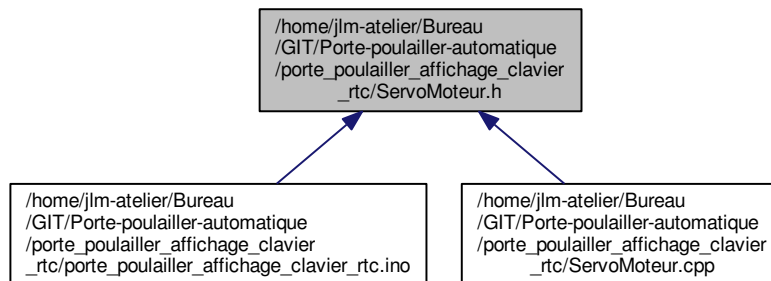
## 6.26 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/porte\_↵ \_poulailler\_affichage\_clavier\_rtc/ServoMoteur.h

```
#include "Arduino.h"
#include <ServoTimer2.h>
```

Graphe des dépendances par inclusion de ServoMoteur.h :



Ce graphe montre quels fichiers incluent directement ou indirectement ce fichier :



### Classes

— class **ServoMoteur**

## 6.27 Référence du fichier /home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/R↵ EADME.md





# Index

/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/README.md, 187	_poulailler_affichage_clavier_rtc/LcdPCF8574.h, 138
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/heuristique porte poulailler.mm, 97	_poulailler_affichage_clavier_rtc/Lumiere.cpp, 139
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/AA_fonctions.cpp, 97	/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/Lumiere.h, 140
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/AA_fonctions.h, 98	/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/PowerTools.cpp, 183
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/Accus.cpp, 128	/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/PowerTools.h, 184
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/Accus.h, 128	/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/Radio.cpp, 184
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/Bouton.cpp, 129	/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/Radio.h, 185
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/Bouton.h, 130	/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/ServoMoteur.cpp, 186
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/Clavier.cpp, 131	/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/ServoMoteur.h, 187
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/Clavier.h, 132	/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/porte_Digole_Serial_I2C_LcdDigoleI2C.h, 137
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/Codeur.cpp, 133	Accus, 10
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/Codeur.h, 133	~Bouton
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/HorlogeDS3232.cpp, 134	Bouton, 16
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/HorlogeDS3232.h, 135	~Clavier
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/HorlogeDS3232.cpp, 136	Clavier, 21
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/HorlogeDS3232.h, 136	Codeur
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/LcdDigoleI2C.cpp, 136	Codeur, 25
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/LcdDigoleI2C.h, 136	~HorlogeDS3232
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/LcdPCF8574.h, 136	HorlogeDS3232, 32
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/LcdPCF8574.h, 136	~LcdDigoleI2C
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/LcdPCF8574.h, 136	LcdDigoleI2C, 45
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/LcdPCF8574.h, 136	LcdPCF8574
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/LcdPCF8574.h, 136	LcdPCF8574, 55
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/LcdPCF8574.h, 136	~Lumiere
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/LcdPCF8574.h, 136	Lumiere, 64
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/LcdPCF8574.h, 136	~PowerTools
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/LcdPCF8574.h, 136	PowerTools, 74
/home/jlm-atelier/Bureau/GIT/Porte-poulailler-automatique/_poulailler_affichage_clavier_rtc/LcdPCF8574.h, 136	Radio

- Radio, 77
- ~ServoMoteur
  - ServoMoteur, 87
- AA\_fonctions.h
  - affiChoixOuvFerm, 99
  - affiFinDeCourseFermeture, 100
  - affiFinDeCourseOuverture, 101
  - affiLumMatin, 101
  - affiLumSoir, 102
  - affiPulsePlusCptRoue, 103
  - affiTensionBatCdes, 103
  - affiTensionBatServo, 104
  - affichageDemarrage, 99
  - choixLumSoir, 105
  - choixOuvFerm, 105
  - closeTime, 106
  - deroulementMenu, 107
  - displayDate, 108
  - displayTime, 108
  - eclairageAfficheur, 109
  - ecritureDateTime, 110
  - enterSleep, 110
  - fermeturePorte, 111
  - ISR, 112
  - lectureClavier, 112
  - lumiere, 113
  - myInterruptINT0, 114
  - myInterruptINT1, 114
  - openTime, 115
  - ouvFermLum, 117
  - ouverturePorte, 116
  - read\_temp, 118
  - regFinDeCourseFermeture, 119
  - regFinDeCourseOuverture, 119
  - reglageDate, 120
  - reglageHeureFermeture, 121
  - reglageHeureOuverture, 121
  - reglageTime, 122
  - routineGestionWatchdog, 122
  - routineInterruptionAlarme2, 123
  - routineInterruptionAlarme1, 124
  - routineInterruptionBp, 124
  - routineTestFermetureBoitier, 125
  - routineTestOuvertureBoitier, 126
  - temporisationAffichage, 126
  - testServo, 127
- Accus, 9
  - ~Accus, 10
  - Accus, 10
  - accusFaible, 10
  - m\_accusPin, 12
  - m\_debug, 12
  - m\_maxCAD, 12
  - m\_rapportConversion, 12
  - m\_tensionMiniAccus, 13
  - tensionAccus, 11
  - tensionAccusCADversFloat, 12
  - tensionAccusCAD, 11
- accusCde
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 175
- accusFaible
  - Accus, 10
- accusPinCde
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 175
- accusPinServo
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 175
- accusServo
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 175
- adresseBoitier24C32
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 175
- affiChoixOuvFerm
  - AA\_fonctions.h, 99
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 144
- affiFinDeCourseFermeture
  - AA\_fonctions.h, 100
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 145
- affiFinDeCourseOuverture
  - AA\_fonctions.h, 101
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 146
- affiLumMatin
  - AA\_fonctions.h, 101
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 146
- affiLumSoir
  - AA\_fonctions.h, 102
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 147
- affiPulsePlusCptRoue
  - AA\_fonctions.h, 103
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 148
- affiTensionBatCdes
  - AA\_fonctions.h, 103
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 148
- affiTensionBatServo
  - AA\_fonctions.h, 104
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 149
- affichageChoix
  - LcdDigoleI2C, 45
  - LcdPCF8574, 56
- affichageDateHeure
  - LcdDigoleI2C, 45
  - LcdPCF8574, 56
- affichageDemarrage
  - AA\_fonctions.h, 99
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 144
- affichageLumFinCourse
  - LcdDigoleI2C, 46
  - LcdPCF8574, 56
- affichageServo
  - LcdDigoleI2C, 46
  - LcdPCF8574, 56
- affichageUneLigne
  - LcdDigoleI2C, 46
  - LcdPCF8574, 57
- affichageVoltage
  - LcdDigoleI2C, 47
  - LcdPCF8574, 58
- alarm\_1

- porte\_poulailler\_affichage\_clavier\_rtc.ino, 175
- alarm\_2
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 175
- annee
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 175
- batterieFaible
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 176
- bcdToDec
  - HorlogeDS3232, 33
- boitierOuvert
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 176
- bonjour
  - LcdDigoleI2C, 48
  - LcdPCF8574, 58
- boucleLumiere
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 176
- boucleTemps
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 176
- bouclesWatchdog
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 176
- Bouton, 13
  - ~Bouton, 16
  - Bouton, 16
  - m\_debounce, 18
  - m\_debug, 18
  - m\_pinBoitier, 18
  - m\_pinBp, 18
  - m\_relacheBp, 18
  - m\_tempoDebounce, 18
  - testBoitierFerme, 16
  - testBoitierOuvert, 16
  - testInterruptionBoitier, 16
  - testInterruptionBp, 17
  - testRelacheBp, 17
  - testToucheBp, 17
- calculTemperature
  - HorlogeDS3232, 33
- chaîneVide
  - Radio, 78
- choixLumMatin
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 150
- choixLumSoir
  - AA\_fonctions.h, 105
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 150
- choixOuvFerm
  - AA\_fonctions.h, 105
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 151
- choixRetroEclairage
  - LcdDigoleI2C, 48
  - LcdPCF8574, 59
- choixTypeOuvertureFermeture
  - HorlogeDS3232, 33
- clav
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 176
- Clavier, 19
  - ~Clavier, 21
  - Clavier, 21
- deplacementDansMenu, 21
- get\_key, 21
- m\_AdcKeyVal, 23
- m\_MenuManuel, 24
- m\_NumKeys, 24
- m\_debug, 23
- m\_oldKey, 24
- positionMenu, 22
- read\_key, 22
- relacheTouche, 23
- testTouche5, 23
- closeTime
  - AA\_fonctions.h, 106
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 152
- codOpt
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 176
- Codeur, 24
  - ~Codeur, 25
  - Codeur, 25
  - compteurRoueCodeuse, 26
  - get\_m\_compteRoueCodeuse, 26
  - get\_m\_finDeCourseFermeture, 26
  - get\_m\_finDeCourseOuverture, 27
  - init, 27
  - m\_compteRoueCodeuse, 29
  - m\_debug, 29
  - m\_finDeCourseFermeture, 29
  - m\_finDeCourseMax, 29
  - m\_finDeCourseOuverture, 29
  - m\_interruptRoueCodeuse, 29
  - m\_positionRoueCodeuse, 29
  - m\_roueCodeusePin, 30
  - reglageFinDeCourse, 27
  - set\_m\_compteRoueCodeuse, 27
  - set\_m\_finDeCourseFermeture, 28
  - set\_m\_finDeCourseOuverture, 28
  - testCompteurRoueCodeuse, 28
- colonnes
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 176
- compteRoueCodeuse
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 176
- compteurRoueCodeuse
  - Codeur, 26
- conversion
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 176
- cursorPosition
  - LcdDigoleI2C, 48
  - LcdPCF8574, 59
- cursorPositionReglages
  - LcdDigoleI2C, 49
  - LcdPCF8574, 60
- DS3231\_I2C\_ADDRESS
  - HorlogeDS3232.h, 136
- debounce
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 176
- debug
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 177
- decToBcd

- HorlogeDS3232, 34
- declenchementServoLuminosite
  - Lumiere, 65
- deplacementDansMenu
  - Clavier, 21
- deroulementMenu
  - AA\_fonctions.h, 107
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 152
- displayDate
  - AA\_fonctions.h, 108
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 154
- displayTime
  - AA\_fonctions.h, 108
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 154
- eclairageAfficheur
  - AA\_fonctions.h, 109
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 155
- ecritureDateTime
  - AA\_fonctions.h, 110
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 155
- effacementAfficheur
  - LcdDigoleI2C, 49
  - LcdPCF8574, 60
- emissionRadio
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 177
- enterSleep
  - AA\_fonctions.h, 110
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 156
- envoiFloat
  - Radio, 78
- envoiMessage
  - Radio, 79
- envoiUnsignedInt
  - Radio, 79
- f\_wdt
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 177
- fenetreNonDeclenchement
  - Lumiere, 65
- fermeturePorte
  - AA\_fonctions.h, 111
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 156
- finDeCourseFermeture
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 177
- finDeCourseOuverture
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 177
- gestionCurseur
  - LcdDigoleI2C, 49
  - LcdPCF8574, 60
- get\_key
  - Clavier, 21
- get\_m\_alarm1Hour
  - HorlogeDS3232, 34
- get\_m\_alarm1Minute
  - HorlogeDS3232, 34
- get\_m\_alarm1Second
  - HorlogeDS3232, 35
- get\_m\_alarm2Hour
  - HorlogeDS3232, 35
- get\_m\_alarm2Minute
  - HorlogeDS3232, 35
- get\_m\_alarm2Hour
  - HorlogeDS3232, 35
- get\_m\_alarm2Minute
  - HorlogeDS3232, 35
- get\_m\_compteRoueCodeuse
  - Codeur, 26
- get\_m\_compteurWatchdogLumiere
  - Lumiere, 65
- get\_m\_decalage
  - LcdDigoleI2C, 50
  - LcdPCF8574, 61
- get\_m\_fermeture
  - Lumiere, 66
- get\_m\_finDeCourseFermeture
  - Codeur, 26
- get\_m\_finDeCourseOuverture
  - Codeur, 27
- get\_m\_lumMatin
  - Lumiere, 66
- get\_m\_lumSoir
  - Lumiere, 66
- get\_m\_ouvFerm
  - ServoMoteur, 88
- get\_m\_ouverture
  - Lumiere, 67
- get\_m\_pulse
  - ServoMoteur, 88
- get\_m\_radio
  - Radio, 80
- get\_m\_servoAction
  - ServoMoteur, 88
- heure
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 177
- heureFenetreSoir
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 177
- HorlogeDS3232, 30
  - ~HorlogeDS3232, 32
  - bcdToDec, 33
  - calculTemperature, 33
  - choixTypeOuvertureFermeture, 33
  - decToBcd, 34
  - get\_m\_alarm1Hour, 34
  - get\_m\_alarm1Minute, 34
  - get\_m\_alarm1Second, 35
  - get\_m\_alarm2Hour, 35
  - get\_m\_alarm2Minute, 35
  - HorlogeDS3232, 32
  - i2c\_eeprom\_read\_byte, 36
  - i2c\_eeprom\_write\_byte, 36
  - init, 36
  - lectureHoraireALARM1, 37
  - lectureHoraireALARM2, 37
  - lectureRegistreEtConversion, 38
  - m\_alarm1Hour, 41
  - m\_alarm1Minute, 41
  - m\_alarm1Second, 41
  - m\_alarm2Hour, 41
  - m\_alarm2Minute, 42

- m\_debug, 42
- m\_deviceAddress, 42
- m\_rtcINT, 42
- reglageAlarme, 39
- reglageHeure, 39
- sauvEepromChoix, 40
- testInterruptRTC, 40
- testPresenceCarterc, 41
- HorlogeDS3232.h
  - DS3231\_I2C\_ADDRESS, 136
- i2c\_eeprom\_read\_byte
  - HorlogeDS3232, 36
- i2c\_eeprom\_write\_byte
  - HorlogeDS3232, 36
- ISR
  - AA\_fonctions.h, 112
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 157
- incrementation
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 177
- init
  - Codeur, 27
  - HorlogeDS3232, 36
  - LcdDigoleI2C, 50
  - LcdPCF8574, 61
  - Radio, 81
  - ServoMoteur, 89
- interruptBp
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 177
- interruptOuvBoi
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 178
- interruptRTC
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 178
- jour
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 178
- jourSemaine
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 178
- LCD\_DIGOLE
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 144
- LED\_PIN
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 144
- LcdCursor
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 178
- LcdDigoleI2C.h
  - \_Digole\_Serial\_I2C\_, 137
- LcdDigoleI2C, 42
  - ~LcdDigoleI2C, 45
  - affichageChoix, 45
  - affichageDateHeure, 45
  - affichageLumFinCourse, 46
  - affichageServo, 46
  - affichageUneLigne, 46
  - affichageVoltage, 47
  - bonjour, 48
  - choixRetroEclairage, 48
  - cursorPosition, 48
  - cursorPositionReglages, 49
  - effacementAfficheur, 49
  - gestionCurseur, 49
  - get\_m\_decalage, 50
  - init, 50
  - LcdDigoleI2C, 45
  - m\_debug, 52
  - m\_decalage, 52
  - m\_deplacement, 52
  - m\_ligne, 52
  - m\_retroEclairage, 52
  - m\_taille, 52
  - razLcd, 50
  - resetPos, 50
  - retroEclairage, 51
  - set\_m\_decalage, 51
  - transformation, 51
- LcdPCF8574, 53
  - ~LcdPCF8574, 55
  - affichageChoix, 56
  - affichageDateHeure, 56
  - affichageLumFinCourse, 56
  - affichageServo, 56
  - affichageUneLigne, 57
  - affichageVoltage, 58
  - bonjour, 58
  - choixRetroEclairage, 59
  - cursorPosition, 59
  - cursorPositionReglages, 60
  - effacementAfficheur, 60
  - gestionCurseur, 60
  - get\_m\_decalage, 61
  - init, 61
  - LcdPCF8574, 55
  - m\_debug, 62
  - m\_decalage, 62
  - m\_deplacement, 62
  - m\_ligne, 62
  - m\_retroEclairage, 62
  - m\_taille, 63
  - razLcd, 61
  - resetPos, 61
  - retroEclairage, 61
  - set\_m\_decalage, 62
  - transformation, 62
- lectureClavier
  - AA\_fonctions.h, 112
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 157
- lectureHoraireALARM1
  - HorlogeDS3232, 37
- lectureHoraireALARM2
  - HorlogeDS3232, 37
- lectureRegistreEtConversion
  - HorlogeDS3232, 38
- loop
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 158
- lum
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 178
- lumMatin

- porte\_poulailler\_affichage\_clavier\_rtc.ino, 178
- lumSoir
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 178
- Lumiere, 63
  - ~Lumiere, 64
  - declenchementServoLuminosite, 65
  - fenetreNonDeclenchement, 65
  - get\_m\_compteurWatchdogLumiere, 65
  - get\_m\_fermeture, 66
  - get\_m\_lumMatin, 66
  - get\_m\_lumSoir, 66
  - get\_m\_ouverture, 67
  - Lumiere, 64
  - luminositeCAD, 67
  - m\_compteurWatchdogLumiere, 72
  - m\_debug, 72
  - m\_fermeture, 72
  - m\_heureFenetreSoir, 72
  - m\_incrementation, 72
  - m\_lumMatin, 72
  - m\_lumSoir, 72
  - m\_lumiereMax, 72
  - m\_lumierePin, 72
  - m\_maxCAD, 73
  - m\_ouverture, 73
  - m\_rapportConversion, 73
  - m\_tempsLum, 73
  - nonDeclenchementPositionServo, 68
  - reglageLumiere, 68
  - set\_m\_compteurWatchdogLumiere, 69
  - set\_m\_fermeture, 69
  - set\_m\_lumMatin, 69
  - set\_m\_lumSoir, 70
  - set\_m\_ouverture, 70
  - tensionLuminosite, 70
  - tensionLuminositeCADversFloat, 71
  - testLuminosite, 71
- lumiere
  - AA\_fonctions.h, 113
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 159
- lumierePin
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 178
- luminositeCAD
  - Lumiere, 67
- m\_AdcKeyVal
  - Clavier, 23
- m\_MenuManuel
  - Clavier, 24
- m\_NumKeys
  - Clavier, 24
- m\_accusPin
  - Accus, 12
- m\_alarm1Hour
  - HorlogeDS3232, 41
- m\_alarm1Minute
  - HorlogeDS3232, 41
- m\_alarm1Second
  - HorlogeDS3232, 41
- m\_alarm2Hour
  - HorlogeDS3232, 41
- m\_alarm2Minute
  - HorlogeDS3232, 42
- m\_chaine
  - Radio, 83
- m\_compteRoueCodeuse
  - Codeur, 29
- m\_compteurWatchdogLumiere
  - Lumiere, 72
- m\_debounce
  - Bouton, 18
- m\_debug
  - Accus, 12
  - Bouton, 18
  - Clavier, 23
  - Codeur, 29
  - HorlogeDS3232, 42
  - LcdDigoleI2C, 52
  - LcdPCF8574, 62
  - Lumiere, 72
  - PowerTools, 76
  - Radio, 83
  - ServoMoteur, 94
- m\_decalage
  - LcdDigoleI2C, 52
  - LcdPCF8574, 62
- m\_deplacement
  - LcdDigoleI2C, 52
  - LcdPCF8574, 62
- m\_deviceAddress
  - HorlogeDS3232, 42
- m\_fermeture
  - Lumiere, 72
- m\_finDeCourseFermeture
  - Codeur, 29
- m\_finDeCourseMax
  - Codeur, 29
- m\_finDeCourseOuverture
  - Codeur, 29
- m\_heureFenetreSoir
  - Lumiere, 72
- m\_incrementation
  - Lumiere, 72
- m\_interruptRoueCodeuse
  - Codeur, 29
- m\_ligne
  - LcdDigoleI2C, 52
  - LcdPCF8574, 62
- m\_lumMatin
  - Lumiere, 72
- m\_lumSoir
  - Lumiere, 72
- m\_lumiereMax
  - Lumiere, 72
- m\_lumierePin
  - Lumiere, 72
- m\_maxCAD

- Accus, 12
- Lumiere, 73
- m\_oldKey
  - Clavier, 24
- m\_ouvFerm
  - ServoMoteur, 94
- m\_ouverture
  - Lumiere, 73
- m\_pinBoitier
  - Bouton, 18
- m\_pinBp
  - Bouton, 18
- m\_pinCde
  - ServoMoteur, 94
- m\_pinEmRadio
  - Radio, 83
- m\_pinRelais
  - ServoMoteur, 94
- m\_pinSecuriteHaute
  - ServoMoteur, 94
- m\_pinSwitchEmissionRadio
  - Radio, 83
- m\_positionRoueCodeuse
  - Codeur, 29
- m\_pulse
  - ServoMoteur, 94
- m\_pulseOuvFerm
  - ServoMoteur, 94
- m\_pulseReduit
  - ServoMoteur, 95
- m\_pulseStop
  - ServoMoteur, 95
- m\_radio
  - Radio, 84
- m\_rapportConversion
  - Accus, 12
  - Lumiere, 73
- m\_relacheBp
  - Bouton, 18
- m\_retroEclairage
  - LcdDigoleI2C, 52
  - LcdPCF8574, 62
- m\_roueCodeusePin
  - Codeur, 30
- m\_rtcINT
  - HorlogeDS3232, 42
- m\_servoAction
  - ServoMoteur, 95
- m\_taille
  - LcdDigoleI2C, 52
  - LcdPCF8574, 63
  - Radio, 84
- m\_tempoDebounce
  - Bouton, 18
- m\_tempsLum
  - Lumiere, 73
- m\_tensionMiniAccus
  - Accus, 13
- m\_vitesseTransmission
  - Radio, 84
- memoireLibre
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 178
- menuChoix
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 179
- menuDate
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 179
- menuFermeture
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 179
- menuFinDeCourseFermeture
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 179
- menuFinDeCourseOuverture
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 179
- menuHeure
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 179
- menuLumiere
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 179
- menuLumiereMatin
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 179
- menuLumiereSoir
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 179
- menuManuel
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 179
- menuOuverture
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 180
- menuTemperature
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 180
- menuTensionBatCdes
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 180
- menuTensionBatServo
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 180
- messageRadio
  - Radio, 81
- messageSansParametre
  - Radio, 82
- minutes
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 180
- modificationVitesse
  - ServoMoteur, 89
- mois
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 180
- monServo
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 180
- myInterruptINT0
  - AA\_fonctions.h, 114
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 160
- myInterruptINT1
  - AA\_fonctions.h, 114
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 160
- nonDeclenchementPositionServo
  - Lumiere, 68
- oldkey
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 180
- openTime
  - AA\_fonctions.h, 115
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 161

- ouvFermLum
  - AA\_fonctions.h, 117
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 163
- ouverturePorte
  - AA\_fonctions.h, 116
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 162
- PROGMEM
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 181
- pinBoitier
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 180
- pinBp
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 180
- pinEmRadio
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 180
- pinSwitchEmissionRadio
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 181
- porte\_poulailler\_affichage\_clavier\_rtc.ino
  - accusCde, 175
  - accusPinCde, 175
  - accusPinServo, 175
  - accusServo, 175
  - adresseBoitier24C32, 175
  - affiChoixOuvFerm, 144
  - affiFinDeCourseFermeture, 145
  - affiFinDeCourseOuverture, 146
  - affiLumMatin, 146
  - affiLumSoir, 147
  - affiPulsePlusCptRoue, 148
  - affiTensionBatCdes, 148
  - affiTensionBatServo, 149
  - affichageDemarrage, 144
  - alarm\_1, 175
  - alarm\_2, 175
  - annee, 175
  - batterieFaible, 176
  - boitierOuvert, 176
  - boucleLumiere, 176
  - boucleTemps, 176
  - bouclesWatchdog, 176
  - choixLumMatin, 150
  - choixLumSoir, 150
  - choixOuvFerm, 151
  - clav, 176
  - closeTime, 152
  - codOpt, 176
  - colonnes, 176
  - compteRoueCodeuse, 176
  - conversion, 176
  - debounce, 176
  - debug, 177
  - deroulementMenu, 152
  - displayDate, 154
  - displayTime, 154
  - eclairageAfficheur, 155
  - ecritureDateTime, 155
  - emissionRadio, 177
  - enterSleep, 156
  - f\_wdt, 177
  - fermeturePorte, 156
  - finDeCourseFermeture, 177
  - finDeCourseOuverture, 177
  - heure, 177
  - heureFenetreSoir, 177
  - ISR, 157
  - incrementation, 177
  - interruptBp, 177
  - interruptOuvBoi, 178
  - interruptRTC, 178
  - jour, 178
  - jourSemaine, 178
  - LCD\_DIGOLE, 144
  - LED\_PIN, 144
  - LcdCursor, 178
  - lectureClavier, 157
  - loop, 158
  - lum, 178
  - lumMatin, 178
  - lumSoir, 178
  - lumiere, 159
  - lumierePin, 178
  - memoireLibre, 178
  - menuChoix, 179
  - menuDate, 179
  - menuFermeture, 179
  - menuFinDeCourseFermeture, 179
  - menuFinDeCourseOuverture, 179
  - menuHeure, 179
  - menuLumiere, 179
  - menuLumiereMatin, 179
  - menuLumiereSoir, 179
  - menuManuel, 179
  - menuOuverture, 180
  - menuTemperature, 180
  - menuTensionBatCdes, 180
  - menuTensionBatServo, 180
  - minutes, 180
  - mois, 180
  - monServo, 180
  - myInterruptINT0, 160
  - myInterruptINT1, 160
  - oldkey, 180
  - openTime, 161
  - ouvFermLum, 163
  - ouverturePorte, 162
  - PROGMEM, 181
  - pinBoitier, 180
  - pinBp, 180
  - pinEmRadio, 180
  - pinSwitchEmissionRadio, 181
  - pulseStop, 181
  - radio, 181
  - rapportConversion, 181
  - read\_temp, 164
  - reduit, 181
  - regFinDeCourseFermeture, 165
  - regFinDeCourseOuverture, 165



- reglage, 181
- reglageDate, 166
- reglageHeureFermeture, 167
- reglageHeureOuverture, 167
- reglageTime, 168
- relache, 181
- roueCodeuse, 181
- routineGestionWatchdog, 168
- routineInterruptionAlarme2, 169
- routineInterruptionAlarme1, 170
- routineInterruptionBp, 170
- routineTestFermetureBoitier, 171
- routineTestOuvertureBoitier, 172
- rtc, 181
- rtcINT, 182
- secondes, 182
- securiteHaute, 182
- sensorClavier, 182
- servoCde, 182
- servoPin, 182
- setup, 172
- temporisationAffichage, 173
- temps, 182
- tempsWatchdog, 182
- tensionMiniAccus, 182
- testServo, 174
- testServoMoteur, 182
- tm, 183
- tools, 183
- touche, 183
- typeTemperature, 183
- vitesseTransmission, 183
- Wire, 183
- positionMenu
  - Clavier, 22
- PowerTools, 73
  - ~PowerTools, 74
  - m\_debug, 76
  - PowerTools, 74
  - setup\_watchdog, 75
  - setupPower, 75
  - tailleChaine, 75
- pulseStop
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 181
- Radio, 76
  - ~Radio, 77
  - chaineVide, 78
  - envoiFloat, 78
  - envoiMessage, 79
  - envoiUnsignedInt, 79
  - get\_m\_radio, 80
  - init, 81
  - m\_chaine, 83
  - m\_debug, 83
  - m\_pinEmRadio, 83
  - m\_pinSwitchEmissionRadio, 83
  - m\_radio, 84
  - m\_taille, 84
  - m\_vitesseTransmission, 84
  - messageRadio, 81
  - messageSansParametre, 82
  - Radio, 77
  - raz\_m\_chaine, 82
  - set\_m\_radio, 82
  - testSwitchEmissionRadio, 83
- radio
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 181
- rapportConversion
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 181
- raz\_m\_chaine
  - Radio, 82
- razLcd
  - LcdDigoleI2C, 50
  - LcdPCF8574, 61
- read\_key
  - Clavier, 22
- read\_temp
  - AA\_fonctions.h, 118
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 164
- reduit
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 181
- regFinDeCourseFermeture
  - AA\_fonctions.h, 119
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 165
- regFinDeCourseOuverture
  - AA\_fonctions.h, 119
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 165
- reglage
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 181
- reglageAlarme
  - HorlogeDS3232, 39
- reglageDate
  - AA\_fonctions.h, 120
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 166
- reglageFinDeCourse
  - Codeur, 27
- reglageHeure
  - HorlogeDS3232, 39
- reglageHeureFermeture
  - AA\_fonctions.h, 121
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 167
- reglageHeureOuverture
  - AA\_fonctions.h, 121
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 167
- reglageLumiere
  - Lumiere, 68
- reglageTime
  - AA\_fonctions.h, 122
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 168
- relache
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 181
- relacheTouche
  - Clavier, 23
- relaisHorsTension
  - ServoMoteur, 90
- relaisSousTension

- ServoMoteur, 90
- resetPos
  - LcdDigolel2C, 50
  - LcdPCF8574, 61
- retroEclairage
  - LcdDigolel2C, 51
  - LcdPCF8574, 61
- roueCodeuse
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 181
- routineGestionWatchdog
  - AA\_fonctions.h, 122
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 168
- routineInterruptionAlarme2
  - AA\_fonctions.h, 123
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 169
- routineInterruptionAlarme1
  - AA\_fonctions.h, 124
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 170
- routineInterruptionBp
  - AA\_fonctions.h, 124
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 170
- routineTestFermetureBoitier
  - AA\_fonctions.h, 125
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 171
- routineTestOuvertureBoitier
  - AA\_fonctions.h, 126
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 172
- rtc
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 181
- rtcINT
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 182
- sauvEepromChoix
  - HorlogeDS3232, 40
- secondes
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 182
- securiteHaute
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 182
- sensorClavier
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 182
- servoCde
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 182
- servoHorsTension
  - ServoMoteur, 91
- ServoMoteur, 84
  - ~ServoMoteur, 87
  - get\_m\_ouvFerm, 88
  - get\_m\_pulse, 88
  - get\_m\_servoAction, 88
  - init, 89
  - m\_debug, 94
  - m\_ouvFerm, 94
  - m\_pinCde, 94
  - m\_pinRelais, 94
  - m\_pinSecuriteHaute, 94
  - m\_pulse, 94
  - m\_pulseOuvFerm, 94
  - m\_pulseReduit, 95
  - m\_pulseStop, 95
  - m\_servoAction, 95
  - modificationVitesse, 89
  - relaisHorsTension, 90
  - relaisSousTension, 90
  - servoHorsTension, 91
  - ServoMoteur, 87
  - servoOuvFerm, 91
  - servoVitesse, 92
  - set\_m\_ouvFerm, 93
  - set\_m\_pulse, 93
  - set\_m\_servoAction, 94
- servoOuvFerm
  - ServoMoteur, 91
- servoPin
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 182
- servoVitesse
  - ServoMoteur, 92
- set\_m\_compteRoueCodeuse
  - Codeur, 27
- set\_m\_compteurWatchdogLumiere
  - Lumiere, 69
- set\_m\_decalage
  - LcdDigolel2C, 51
  - LcdPCF8574, 62
- set\_m\_fermeture
  - Lumiere, 69
- set\_m\_finDeCourseFermeture
  - Codeur, 28
- set\_m\_finDeCourseOuverture
  - Codeur, 28
- set\_m\_lumMatin
  - Lumiere, 69
- set\_m\_lumSoir
  - Lumiere, 70
- set\_m\_ouvFerm
  - ServoMoteur, 93
- set\_m\_ouverture
  - Lumiere, 70
- set\_m\_pulse
  - ServoMoteur, 93
- set\_m\_radio
  - Radio, 82
- set\_m\_servoAction
  - ServoMoteur, 94
- setup
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 172
- setup\_watchdog
  - PowerTools, 75
- setupPower
  - PowerTools, 75
- tailleChaine
  - PowerTools, 75
- temporisationAffichage
  - AA\_fonctions.h, 126
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 173
- temps
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 182
- tempsWatchdog

- porte\_poulailler\_affichage\_clavier\_rtc.ino, 182
- tensionAccus
  - Accus, 11
- tensionAccusCADversFloat
  - Accus, 12
- tensionAccusCAD
  - Accus, 11
- tensionLuminosite
  - Lumiere, 70
- tensionLuminositeCADversFloat
  - Lumiere, 71
- tensionMiniAccus
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 182
- testBoitierFerme
  - Bouton, 16
- testBoitierOuvert
  - Bouton, 16
- testCompteurRoueCodeuse
  - Codeur, 28
- testInterruptRTC
  - HorlogeDS3232, 40
- testInterruptionBoitier
  - Bouton, 16
- testInterruptionBp
  - Bouton, 17
- testLuminosite
  - Lumiere, 71
- testPresenceCarteRTC
  - HorlogeDS3232, 41
- testRelacheBp
  - Bouton, 17
- testServo
  - AA\_fonctions.h, 127
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 174
- testServoMoteur
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 182
- testSwitchEmissionRadio
  - Radio, 83
- testTouche5
  - Clavier, 23
- testToucheBp
  - Bouton, 17
- tm
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 183
- tools
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 183
- touche
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 183
- transformation
  - LcdDigoleI2C, 51
  - LcdPCF8574, 62
- typeTemperature
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 183
- vitesseTransmission
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 183
- Wire
  - porte\_poulailler\_affichage\_clavier\_rtc.ino, 183