001 //Гиль Никита Александрович

002 //группа 150501

003 //Микропроцессорное устройство контроля

004 //параметров велосипеда

005

006 //библиотека для ИК приемника

007 #include <NecDecoder.h>

008 //библиотека для датчика температура

009 #include <Adafruit\_AHTX0.h>

010 //библиотека для lcd дисплея

011 #include <LiquidCrystal\_I2C.h>

012 //библиотека для gps датчика

013 #include <TinyGPS++.h>

014 //библиотека для порта передачи с gps датчиком

015 #include <SoftwareSerial.h>

016 //библиотека для часов реального времени

017 #include <iarduino\_RTC.h>

018 //библиотека для EEPROM

019 #include <EEPROM.h>

020

021 #define \_RX 0 //вывод RX ардуино

022 #define \_TX 1 //вывод TX ардуино

023 #define \_PIN\_13 13 //вывод D13 ардуино

024 #define \_PIN\_12 12 //вывод D12 ардуино

025 #define \_PIN\_11 11 //вывод D11 ардуино

026 #define \_PIN\_10 10 //вывод D10 ардуино

027 #define \_PIN\_8 8 //вывод D8 ардуино

028 #define \_PIN\_4 4 //вывод D4 ардуино

029 #define \_PIN\_3 3 //вывод D3 ардуино

030

031 #define \_INT0 0 //прерывание 0 ардуино

032 #define \_INT1 1 //прерывание 1 ардуино

033

034 //скорость передачи порта gps

035 #define speedGPSPort 9600

036

037 //количество стартовых сигналов

038 #define startSygnals 3

039 //стартовая задержка

040 #define startDelay 400

041

042 //стартовая скорость ограничения

043 #define startControlSpeed 25

044 //нижняя граница скорости ограничения

045 #define lowSpeed 10

046 //верхняя граница скорости ограничения

047 #define highSpeed 40

048

049 //размеры колес

050 #define size26 "26"

051 #define size27\_5 "27.5"

052 #define size29 "29"

053

054 //длины окружности колес

055 #define length26 2.1

056 #define length27\_5 2.2

057 #define length29 2.3

058

059 //значения для вывода информации

060 #define time 0 //вывод времени

061 #define speedDir 1 //вывод скорости и направления

062 #define speedParams 2 //вывод параметров скорости

063 #define tempHum 3 //вывод температуры и влажности

064 #define distance 4 //вывод расстояний

065

066 //окно коллизий

067 #define windowCollision 80

068

069 //адреса данных в EEPROM

070 #define addressAllDist 0 //адрес всего расстояния

071 #define addressTodayDist 8 //адрес сегоднешнего расстояния

072 #define addressDay 16 //адрес дня последнего измерения

073 #define addressMonth 24 //адрес месяца последнего измерения

074

075 //задержки

076 #define delay1sec 1000 // 1 секунда

077 #define delay2sec 2000 // 2 секунда

078 #define delay4sec 4000 // 4 секунда

079

080 //частота пьезодинамика

081 #define freq\_2000 2000

082

083 //количество цифр в числе

084 #define have1Digit 10 //1 цифра

085 #define have2Digit 100 //2 цифры

086

087 //коды кнопок

088 #define button\_0 48 //обнуление расстояний

089 #define button\_1 48 //выбор 26 размера колеса

090 #define button\_2 24 //выбор 27.5 размера колеса

091 #define button\_3 122 //выбор 29 размера колеса

092 #define button\_minus 224 //уменьшение скорости контроля на 1

093 #define button\_plus 168 //увеличение скорости контроля на 1

094 #define button\_pause 194 //включить/выключить контроль скорости

095

096 #define button\_ch1 2 //вывод времени

097 #define button\_ch2 98 //вывод скорости и направления

098 #define button\_ch3 34 //вывод параметров скорости

099 #define button\_ch4 162 //вывод температуры и влажности

100 #define button\_ch5 226 //вывод расстояний

101

102 //структура времени

103 typedef struct timeStruct {

104 int hour; //часы

105 int min; //минуты

106 int sec; //секунды

107 };

108

109 //структура для хранения времени

110 timeStruct tm;

111

112 //gps датчик

113 TinyGPSPlus gps;

114 //порт для соединения к gps

115 SoftwareSerial SoftSerial(\_RX, \_TX);

116

117 //lcd дисплей

118 LiquidCrystal\_I2C lcd(0x27, 20, 4);

119

120 //часы реального времени

121 iarduino\_RTC clock(RTC\_DS1302, \_PIN\_12, \_PIN\_8, \_PIN\_10);

122

123 //ИК приемник

124 NecDecoder ir;

125

126 //датчик температуры

127 Adafruit\_AHTX0 aht;

128 //значения измерений

129 sensors\_event\_t humidity, temp;

130

131 //скорость ограничения

132 int speedControl = startControlSpeed;

133 //флаг контроля скорости

134 volatile bool flagControlSpeed = true;

135 //скорость

136 int speedKmh;

137

138 //последнее измерение скорости

139 unsigned long lastFlash;

140

141 //последнее снятие значения с датчика температуры

142 unsigned long takeInfoAht;

143

144 //последняя запись в EEPROM и получение информации от gps

145 unsigned long writing;

146

147 //текущая длина окружности колеса

148 volatile float nowLength = length26;

149

150 //флаг для вывода информации

151 volatile int flagOut = 0;

152 //значение до изменения флага вывода

153 int lastFlagOut = 0;

154

155 //значение времени

156 unsigned long mill;

157

158 //расстояние за поездку

159 volatile double travelDist = 0;

160 //все расстояние

161 volatile double allDist = 0;

162 //расстояние за сегодня

163 volatile double todayDist = 0;

164

165 int day; //день

166 int month; //месяц

167

168 //флаг очистки расстояний

169 bool flagClearDist = false;

170

171 //функция перевода секунд во время

172 void secondsToTime(unsigned long seconds) {

173 tm.sec = seconds % 60; //получение секунд

174 tm.min = (seconds / 60) % 60; //получение минут

175 tm.hour = (seconds / 3600) % 24; //получение часов

176 }

177

178 //функция обработки нажатия клавиш

179 void irIsr() {

180 ir.tick(); //начать обработку

181 //если доступно

182 if (ir.available()) {

183 //чтение значения кнопки

184 int value = ir.readCommand();

185

186 //если нажата кнопка 1 и установлен флаг вывода параметров скорости

187 if(value == button\_1 && flagOut == speedParams) {

188 nowLength = length26; //изменить размер колеса

189 }

190 //если нажата кнопка 2 и установлен флаг вывода параметров скорости

191 if(value == button\_2 && flagOut == speedParams) {

192 nowLength = length27\_5; //изменить размер колеса

193 }

194 //если нажата кнопка 3 и установлен флаг вывода параметров скорости

195 if(value == button\_3 && flagOut == speedParams) {

196 nowLength = length29; //изменить размер колеса

197 }

198

199 //если нажата кнопка минус, контролируемая скорость больше минимальной

200 // и установлен флаг вывода параметров скорости

201 if(value == button\_minus && speedControl > lowSpeed

202 && flagOut == speedParams) {

203 speedControl--; //уменьшение контролируемой скорости

204 }

205 //если нажата кнопка плюс, контролируемая скорость меньше максимальной

206 // и установлен флаг вывода параметров скорости

207 if(value == button\_plus && speedControl < highSpeed

208 && flagOut == speedParams) {

209 speedControl++; //увеличение контролируемой скорости

210 }

211

212 //если нажата кнопка канал1

213 if(value == button\_ch1) {

214 flagOut = speedDir; //установить вывод скорости и направления

215 }

216 //если нажата кнопка канал2

217 if(value == button\_ch2) {

218 flagOut = tempHum; //установить вывод температуры и влажности

219 }

220 //если нажата кнопка канал3

221 if(value == button\_ch3) {

222 flagOut = speedParams;//установить вывод параметров скорости

223 }

224 //если нажата кнопка канал4

225 if(value == button\_ch4) {

226 flagOut = time; //установить вывод времени

227 }

228 //если нажата кнопка канал5

229 if(value == button\_ch5) {

230 flagOut = distance; //установить вывод расстояний

231 }

232

233 //если нажата кнопка 0 и флаг вывода установлен на расстояние

234 if(value == button\_0 && flagOut == distance) {

235 //обнуление всех расстояний

236 allDist = 0;

237 todayDist = 0;

238 travelDist = 0;

239 //установка флага обнуления расстояний

240 flagClearDist = true;

241 }

242

243 //если нажата кнопка пауза и установлен флаг вывода

244 //параметров скорости

245 if(value == button\_pause && flagOut == speedParams) {

246 if(flagControlSpeed){

247 flagControlSpeed = false; //сброс флага котроля скорости

248 } else {

249 flagControlSpeed = true; //установка флага котроля скорости

250 }

251 }

252 }

253 }

254

255 //функция измерения скорости

256 void speed() {

257 //сохранение времени измерения

258 mill = millis();

259 //ожидание окна коллизий

260 if(mill - lastFlash > windowCollision){

261 //вычисление скорости

262 speedKmh = (nowLength / (((float)(mill - lastFlash)) / 1000)) \* 3.6;

263 //сохранение последнего времени измерения

264 lastFlash = mill;

265 //увеличение расстояний

266 travelDist += nowLength;

267 allDist += nowLength;

268 todayDist += nowLength;

269 }

270 }

271

272 //функция вывода температуры и влажности

273 void outTempHum() {

274 //установка курсора в позицию 0 0

275 lcd.setCursor(0, 0);

276 //вывод температуры

277 lcd.print("Temperature: ");

278 lcd.print((int)temp.temperature);

279 lcd.print(" C");

280 //установка курсора в позицию 1 1

281 lcd.setCursor(0, 1);

282 //вывод влажности

283 lcd.print("Humidity: ");

284 lcd.print((int)humidity.relative\_humidity);

285 lcd.print(" %");

286 }

287

288 //функция вывода скорости и направления

289 void outSpeedDir() {

290 //установка курсора в позицию 0 0

291 lcd.setCursor(0, 0);

292 //вывод скорости

293 lcd.print("SPEED (km/h): ");

294 lcd.print(speedKmh);

295 lcd.print(" ");

296 //установка курсора в позицию 0 1

297 lcd.setCursor(0, 1);

298 lcd.print("DIRACTION");

299 //установка курсора в позицию 0 2

300 lcd.setCursor(0, 2);

301 lcd.print("degrees: ");

302 //если курс определен

303 if(gps.course.isValid()) {

304 //получение градусов

305 int deg = gps.course.deg();

306 //вывод градусов

307 lcd.print(deg);

308 //если цифр меньше 2

309 if(deg < have2Digit) {

310 lcd.print(" "); //удаление оставшейся цифры

311 }

312 //если цифр меньше 1

313 if(deg < have1Digit) {

314 lcd.print(" "); //удаление оставшейся цифры

315 }

316 }

317 }

318

319 //функция вывод параметров скорости

320 void outSpeedParams() {

321 //установка курсора в позицию 0 0

322 lcd.setCursor(0, 0);

323 lcd.print("SIZE: ");

324 //если размер 26 дюймов

325 if(nowLength == length26){

326 lcd.print(size26); //вывод размера

327 lcd.print(" ");

328 //если размер 27.5 дюймов

329 } else if(nowLength == length27\_5) {

330 lcd.print(size27\_5); //вывод размера

331 //если размер 29 дюймов

332 } else {

333 lcd.print(size29); //вывод размера

334 lcd.print(" ");

335 }

336 //установка курсора в позицию 0 1

337 lcd.setCursor(0, 1);

338 lcd.print("CONTROL: ");

339 //если скорость контролируется

340 if(flagControlSpeed) {

341 lcd.print("ON"); //вывод сообщения "включено"

342 lcd.print(" ");

343 //если скорость не контролируется

344 } else {

345 lcd.print("OFF"); //вывод сообщения "отключено"

346 }

347 //установка курсора в позицию 0 2

348 lcd.setCursor(0, 2);

349 //вывод контролируемой скорости

350 lcd.print("CONTROL SPEED: ");

351 lcd.print(speedControl);

352 }

353

354 //функция вывода времени

355 void outTime() {

356 //установка курсора в позицию 0 0

357 lcd.setCursor(0, 0);

358 //вывод текущей даты

359 lcd.print("DATE: ");

360 lcd.print(clock.gettime("d-m-Y"));

361 //установка курсора в позицию 0 1

362 lcd.setCursor(0, 1);

363 //вывод текущего времени

364 lcd.print("TIME: ");

365 lcd.print(clock.gettime("H:i:s"));

366 //установка курсора в позицию 0 2

367 lcd.setCursor(0, 2);

368 //вывод времени поездки

369 lcd.print("TRAVEL: ");

370 secondsToTime(millis() / 1000);

371 //если число состоит 1 цифры

372 if(tm.hour < have1Digit) {

373 lcd.print("0");

374 }

375 //вывод часов

376 lcd.print(tm.hour);

377 lcd.print(":");

378 //если число состоит 1 цифры

379 if(tm.min < have1Digit) {

380 lcd.print("0");

381 }

382 //вывод минут

383 lcd.print(tm.min);

384 lcd.print(":");

385 //если число состоит 1 цифры

386 if(tm.sec < have1Digit) {

387 lcd.print("0");

388 }

389 //вывод секунд

390 lcd.print(tm.sec);

391 }

392

393 //вывод расстояний

394 void outDistance() {

395 //если установлен флаг очистки

396 if(flagClearDist) {

397 flagClearDist = false; //сброс флага очистки

398 lcd.clear(); //очистка экрана

399 }

400 //установка курсора в позицию 0 0

401 lcd.setCursor(0, 0);

402 lcd.print("DISTANCE (m)");

403 //установка курсора в позицию 0 1

404 lcd.setCursor(0, 1);

405 //вывод расстояния за все время

406 lcd.print("ALL TIME: ");

407 lcd.print((long)allDist);

408 //установка курсора в позицию 0 2

409 lcd.setCursor(0, 2);

410 //вывод расстояния за сегодня

411 lcd.print("TODAY: ");

412 lcd.print((long)todayDist);

413 //установка курсора в позицию 0 3

414 lcd.setCursor(0, 3);

415 //вывод расстояния за поездку

416 lcd.print("NOW: ");

417 lcd.print((long)travelDist);

418 }

419

420 //установка начальных значений

421 void setup() {

422 //включение порта GPS

423 SoftSerial.begin(speedGPSPort);

424

425 //установка выводов на выход

426 pinMode(\_PIN\_13, OUTPUT);

427 pinMode(\_PIN\_11, OUTPUT);

428 pinMode(\_PIN\_4, OUTPUT);

429

430 //установка 0 прерывания (ик приемник)

431 attachInterrupt(\_INT0, irIsr, FALLING);

432

433 //установка 1 прерывания (герконовый датчик)

434 attachInterrupt(\_INT1, speed, RISING);

435

436 //запуск работы с датчиком температуры

437 aht.begin();

438

439 //запуск дисплея

440 lcd.init();

441 lcd.backlight();

442

443 //цикл сигнализирования о запуске устройства

444 for(int i = 0; i < startSygnals; ++i){

445 //включение диода

446 digitalWrite(\_PIN\_13, HIGH);

447 //задержка

448 delay(startDelay);

449 //выключение диода

450 digitalWrite(\_PIN\_13, LOW);

451 //задержка

452 delay(startDelay);

453 }

454

455 //запуск работы с часами реального времени

456 clock.begin();

457 //установка времени (при необходимости расскомментировать)

458 //clock.settime(0, 14, 18, 21, 11, 23, 2);

459 //получение из памяти всего пути

460 EEPROM.get(addressAllDist, allDist);

461 //получение из памяти пути за день

462 EEPROM.get(addressTodayDist, todayDist);

463 //получение дня последнего запуска

464 EEPROM.get(addressDay, day);

465 //получение месяца последнего запуска

466 EEPROM.get(addressMonth, month);

467 //если дни не совпадают

468 if(day != clock.day || month != clock.month) {

469 //обнуление расстояния за сегодня

470 todayDist = 0;

471 //запись в память расстояния, дня и месяца

472 EEPROM.put(addressTodayDist, todayDist);

473 EEPROM.put(addressDay, (int)clock.day);

474 EEPROM.put(addressMonth, (int)clock.month);

475 }

476 }

477

478 //цикл выполнения программы

479 void loop() {

480 //если пришло время записи в память и чтения из порта gps

481 if(millis() - writing > delay1sec) {

482 writing = millis(); //сохрание времени последней записи

483 //сохранение всего расстояния

484 EEPROM.put(addressAllDist, allDist);

485 //сохранение расстояния за сегодня

486 EEPROM.put(addressTodayDist, todayDist);

487 //если порт доступен

488 if(SoftSerial.available() > 0) {

489 //декодирование данных

490 gps.encode(SoftSerial.read());

491 }

492 }

493

494 //если долгое время не было измерений

495 if (millis() - lastFlash > delay2sec) {

496 speedKmh = 0; //обнуление скорости

497 }

498

499 //если пришло время получить вданные с датчика температуры

500 if(millis() - takeInfoAht > delay4sec) {

501 //сохранение времени обращения к датчику температуры

502 takeInfoAht = millis();

503 //получение значений с датчика

504 aht.getEvent(&humidity, &temp);

505 }

506

507 //фиксация превышения скорости

508 if(speedKmh > speedControl && flagControlSpeed) {

509 digitalWrite(\_PIN\_11, HIGH); //открытие реле

510 tone(\_PIN\_4, freq\_2000); //запустили звучание

511 } else {

512 digitalWrite(\_PIN\_11, LOW); //закрытие реле

513 noTone(\_PIN\_4); //остановили звучание

514 }

515

516 //если флаг вывода был изменен

517 if(flagOut != lastFlagOut) {

518 lcd.clear(); //очистка экрана

519 //сохранение последнего значения флага

520 lastFlagOut = flagOut;

521 }

522

523 //если флаг установлен в вывод параметров скорости

524 if(flagOut == speedParams) {

525 outSpeedParams(); //вывод параметров скорости

526 }

527

528 //если флаг установлен в вывод скорости и направления

529 if(flagOut == speedDir) {

530 outSpeedDir(); //вывод скорости и направления

531 }

532

533 //если флаг установлен в вывод температуры и влажности

534 if(flagOut == tempHum) {

535 outTempHum(); //вывод температуры и влажности

536 }

537

538 //если флаг установлен в вывод времени

539 if(flagOut == time) {

540 outTime(); //вывод времени

541 }

542

543 //если флаг установлен в вывод расстояния

544 if(flagOut == distance) {

545 outDistance(); //вывод расстояния

546 }

547 }

548