



الجمهورية العربية السورية

جامعة تشرين

كلية الهندسة الميكانيكية والكهربائية

قسم هندسة الاتصالات والالكترونيات

دراسة وتصميم عقد شبكة حساسات لاسلكية بأسلوب Hardware-In-The-Loop

(مشروع اعد لنيل شهادة البكالوريوس في هندسة الاتصالات والالكترونيات)

إعداد الطلاب

الطالب

فارس عزيز الحامض

الطالب

عمّار ياسر رجب

بإشراف الدكتور المهندس

غادي محمودي

الغاية من المشروع

تصميم وتنفيذ شبكة حساسات لاسلكية ومحاكاة عمل هذه الشبكة بتقنية Hardware in the loop.

في هذه التقنية يتم ربط مجموعة عقد حساسات لاسلكية حقيقية بشبكة من عقد الحساسات اللاسلكية الافتراضية والتي تمت محاكاتها باستخدام برنامج ++OMENT بحيث تشكل كل هذه العقد شبكة حساسات لاسلكية كبيرة والتي يتم فيها تبادل الرزم بين العقد الحقيقية والعقد الوهمية.

تسمح هذه الطريقة باختبار أداء شبكة كبيرة من العقد دون الحاجة الى وجود كل عقد هذه الشبكة على ارض الواقع, حيث يتم كشف عيوب العقد الحقيقية والاختفاء في برنامج متحكمات هذه العقد مثل أخطاء اخذ القراءة من المحول التشابهي الرقمي و تحويلها الى قيمة مقروءة , واخطاء التحميل الزائد للعقد, كما تساعد هذه الطريقة في كشف أداء أجهزة الارسال استقبال من عرض حزمة و استهلاك الطاقة, كما يتم أيضا و من خلال وجود عدد كبير من العقد (الحقيقية و الافتراضية) كشف عيوب البروتوكولات المستخدمة في الشبكة كبروتوكولات التوجيه بحيث يمكن تغيير بارامترات العقد الوهمية بسرعة ويسر عند الحاجة في حين ستتطلب هذه العملية وقتا وجهدا كبيرين في حال كان الاختبار يتم على العقد الحقيقية فقط وبالتالي تم توفير الكثير من الوقت والجهد.

الفهرست

1	الغاية من المشروع
4	المقدمة
6	الفصل الأول: أقسام المشروع
6	1-1 محاكي الشبكات OMNET++
6	1-1-1 تصميم العقد ضمن المحاكى
6	2-1-1 محاكاة بروتوكول التوجيه ضمن المحاكى
	3-1-1 تعامل المحاكى مع برنامج خارجي في الزمن الحقيقي من خلال الصنف
8	:SOCKETRTSCHEDULER
9	2-1 تصميم العقد الحقيقية
9	1-2-1 المتحكم PIC12LF1552
10	3-1 دارة الربط بين البيئة الحقيقية والبيئة الافتراضية
12	الفصل الثاني: الواجهات البرمجية للبيئة الافتراضية
12	1-2 طبقة الشبكة NETWORK LAYER
13	2-2 طبقة التطبيقات APPLICATION LAYER
14	الفصل الثالث: الواجهات البرمجية للبيئة الحقيقية
14	1-3 مكتبة NRF24L01
14	2-3 مكتبة NRF_TOOLS
15	3-3 الطبقة الفيزيائية PHYSICAL LAYER
16	4-3 طبقة نقل المعطيات DATALINK LAYER
16	5-3 طبقة الشبكة NETWORK LAYER
17	6-3 طبقة التطبيقات

18	الفصل الرابع: دارة الربط بين البيئتين الافتراضية والحقيقية
18	1-4 عناصر الدارة
18	1-1-4 المتحكم PIC16F877A
18	2-1-4 جهاز الارسال والاستقبال NRF24L01
20	3-1-4 دارة الملائمة MAX232
20	2-4 الواجهة البرمجية لدارة الربط
23	الفصل الخامس: مخططات الشبكة
23	1-5 مخطط الشبكة في المحاي
23	2-5 الية عمل الشبكة
27	الفصل السادس: تطويرات مستقبلية
27	1-5 تعدد البروتوكولات في المحاكاة
27	2-5 انترنت الاشياء
28	قائمة بالملاحق
29	الملحق (أ): بروتوكول الوصل التسلسلي SPI
36	الملحق (ب): جدول تعليمات التحكم بجهاز الارسال والاستقبال عبر الحاسب من خلال دارة الربط
38	الملحق (ج): بروتوكول ENHANCED SHOCKBURST
50	قائمة بالمصطلحات العلمية
52	قائمة بالجداول الواردة
53	قائمة بالأشكال الواردة
55	المراجع العلمية المستخدمة

المقدمة

إن حاجة الانسان الى مراقبة البيئة المحيطة دفعته الى تصميم واختراع الحساسات والتي هي اجهزة تستقبل منها وتستجيب له ويختلف شكل ومستوى تعقيد الحساسات بحسب الغاية من استخدامها والدقة المطلوبة منها، وقد تطورت الحساسات مع التقدم التكنولوجي في العصر الحديث واصبحت معظمها تستجيب للمنبهات بإعطاء اشارة كهربائية تعبر عن شدة المنبه، ومع زيادة الحاجة للحساسات مثل مراقبة مناطق واسعة كما في مراقبة الغابات للحد من الحرائق او مراقبة الحدود بين الدول او حتى مراقبة مداخل المباني أو درجة حرارة مجموعة اجهزة في منشأة صناعية ما، ظهرت الحاجة الملحة لشبكات الحساسات اللاسلكية التي تختصر التكلفة وتتمكن من تغطية مساحات كبيرة و مراقبة المناطق التي لا يمكن الوصول اليها. لكن وباعتبار ان اختبار عمل عدد كبير من الحساسات اللاسلكية ضمن شبكة يعتبر عملية معقدة ومكلفة ماديا وزمنيا إذا تم استخدام أجهزة حقيقية بشكل مباشر اثناء تصميم الشبكة، ويعتبر عملية غير دقيقة إذا تم استخدام المحاكاة فقط، لذا قمنا في هذا المشروع بمحاكاة عمل شبكة كبيرة من العقد اللاسلكية بتكلفة لا تتجاوز سعر بضعة من هذه العقد، وبدقة مقبولة، وبطريقة مرنة تسمح بالتعديل على تصميم الشبكة بشكل سهل وبسيط. عملية المحاكاة هذه تتيح تتبع الرزم بين العقد وفي حال حدوث فشل في الشبكة يمكن الكشف عنه ومعرفة أسبابه بسرعة وسهولة مما يختصر الكثير من الوقت والجهد. بالإضافة الى عملية المحاكاة يمكن اتاحة الوصول الى شبكة الحساسات اللاسلكية عبر الانترنت بحيث تتوفر معلومات عن الشبكة ويمكن الوصول الى معلومات أي عقدة فيها وذلك يعرف بإنترنت الاشياء (Internet Of Things (IOT. أي مثلا يتم ربط شبكة الحساسات اللاسلكية مع موقع الكتروني يقوم بعرض القراءات الخاصة بكل حساس. عملية المحاكاة تمت بإنشاء شبكة حساسات لاسلكية مؤلفة من بضعة عقد حقيقية ومجموعة كبيرة من العقد الافتراضية، وهذه العقد (الحقيقية والافتراضية) مرتبطة مع بعضها وتتبادل الرزم فيما بينها هذه التقنية تعرف بـ Hardware-in-the-loop simulation وهي تقنية مستخدمة بشكل واسع في تطوير واختبار عمل الأنظمة المعقدة في الزمن الحقيقي، وتستخدم تقنية المحاكاة هذه في الحالات التي تتطلب تحسين كفاءة عملية اختبار الأنظمة، وفي حالات محدودية الجدول الزمني الخاص بعملية تطوير نظام ما، وفي حالات الأنظمة ذات الكلفة الاقتصادية العالية، وفي الحالات التي تتطلب معايير بشرية أثناء تصميم النظام. باعتبار عملية المحاكاة هذه تتضمن قسمين (افتراضي وحقيقي) فإن الصعوبات الأساسية التي ظهرت اثناء تنفيذ المشروع عديدة منها:

- صعوبة عملية الاختبار والتحقق من الأخطاء البرمجية في جهاز الارسال والاستقبال، وذلك بسبب عدم إمكانية الوصل المباشر مع الحاسب او مع أي جهاز اظهر آخر بالتالي لإدارة عمل هذا الجهاز قمنا بكتابة كود برمجي يعمل على متحكم موصول بجهاز الارسال والاستقبال من جهة وبالحاسب من جهة أخرى يعمل كواجهة ربط بينهما.
- ظهور مشاكل تتعلق بتأخر زمن المحاكاة عن الزمن الحقيقي، وذلك بسبب اختلاف معدلات نقل البيانات بين الأجهزة المختلفة في المشروع، وبسبب ضعف أداء الحاسب (مع زيادة عدد العقد الافتراضية تزداد الحمولة على المعالج ويزداد التأخير). تم تجاوز هذه المشاكل من خلال تحقيق التقارب بين معدلات نقل البيانات لتقنيات الربط المختلفة واستخدام حاسب يتمتع بسرعة معالجة أكبر.
- إيجاد التصميم المناسب الذي يضمن عمل شبكة الحساسات اللاسلكية بعقدها الحقيقية والافتراضية دون أن تأخذ هذه العقد بعين الاعتبار اختلاف الواجهات والأجهزة اللازمة لربط البيئة الافتراضية مع البيئة الحقيقية.

يتحدث الفصل الأول عن أقسام المشروع بشكل عام حيث يتناول لمحة عامة عن المحاكى المستخدم، وكيفية تصميم العقد ضمنه، كما يتحدث عن بروتوكول التوجيه وكيفية محاكاته ضمن برنامج المحاكاة، ويبين بشكل عام آلية الربط بين البيئتين الحقيقية والافتراضية وكيفية تصميم العقدة الحقيقية. بينما يبين الفصل الثاني الواجهة البرمجية للبيئة الافتراضية والتوابع المستخدمة في طبقات الشبكة والتطبيقات، أما الفصل الثالث فيتحدث عن الواجهة البرمجية للعقد الحقيقية، والفصل الرابع يتناول الحديث عن دارة الربط من عناصر الكترونية وأكواد برمجية. والفصل الخامس يحوي مخططا لشبكة الحساسات اللاسلكية التي قمنا بمحاكاتها إضافة الى شرح مبدأ العمل، أما التطويرات المستقبلية للمشروع والتحسينات التي يمكن اضافتها على هذا المشروع فيتحدث عنها الفصل السادس.

الفصل الأول:

أقسام المشروع

1-1 محاكي الشبكات OMNET++

يقوم هذا البرنامج بمحاكاة عمل الشبكات بشكل عام، حيث يمكن إضافة وبرمجة وحدات بسيطة (Simple module) وبناء وحدات مركبة (Compound modules) لبناء الشبكة. لبرمجة وحدات بسيطة في المحاكى يتم تحديد بوابات دخل وخرج لكل وحدة كما يتم برمجة الوحدة من خلال لغة الـ C++ والمكتبات الخاصة بالمحاكي، كما يمكن وصل وحدات بسيطة معا عبر البوابات لبناء وحدات مركبة. ان التتابع الأساسية التي يجب تعريفها في برمجة الوحدة البسيطة هي:

```
void initialize() -
void handleMessage(cMessage *msg) -
```

التابع initialize يتم استدعاؤه مرة واحد في بداية المحاكاة وذلك لكل وحدة، لذا من المفترض ان يتم تهيئة الوحدة البسيطة ضمن هذا التابع. أما التابع handleMessage يتم استدعاؤه عند حصول حدث ضمن الوحدة، وهذا الحدث قد يكون رزمة خارجية من وحدة أخرى أو حدث محلي أو أنواع أخرى من الاحداث. من ميزات هذا المحاكى هو إمكانية تعريف رزمة Packet يمكن من خلالها محاكاة حقول بروتوكول ما.

1-1-1 تصميم العقد ضمن المحاكى

في الشبكة التي قمنا بتصميمها باستخدام محاكي OMNET++، جعلنا العقدة اللاسلكية الافتراضية عبارة عن وحدة بسيطة، وفي تابع التهيئة initialize لهذه الوحدة يتم اختيار نوع بروتوكول التوجيه كما يتم تحديد عنوان العقدة، ويتم أيضا جدولة عملية أخذ قراءة الحساس (توليد قيم عشوائية على أنها قراءة الحساس).

التابع handleMessage(cMessage *msg) في العقدة الافتراضية يتم فيه التعامل مع الرزم الخارجية بحسب بروتوكول التوجيه، ومع الاحداث المحلية (والتي هي في هذه الحالة عمليات توليد القراءات) يتم عندها ارسال رسالة تحوي القراءة وذلك بحسب بروتوكول التوجيه المستخدم.

1-1-2 محاكاة بروتوكول التوجيه ضمن المحاكى

قمنا في برنامج المحاكاة بتعريف شكل للرزمة وذلك بحيث تحقق آلية عمل بروتوكول التوجيه Sequential Assignment Routing (SAR)، حيث تتألف الرزمة من الحقول التالية:

جدول 1: حقول الرزمة في بروتوكول التوجيه SAR

الوصف	حجم المتحول	نوع المتحول	اسم المتحول
عنوان المرسل	1 Byte	unsigned char	sndAddress
عنوان الهدف	1 Byte	unsigned char	dstAddress
عدد القفزات	1 Byte	unsigned char	hopCount
عدد القيم (عدد الرسائل)	1 Byte	unsigned char	msgCount
البيانات (القراءات)	4*5 Byte	float	data[5]
مصدر البيانات (القراءات)	1*5 Byte	unsigned char	dataSrc[5]

ان حقل البيانات data[5] هو عبارة عن مصفوفة عناصرها من النوع float وتحتوي 5 مواقع (تسمح بتخزين 5 قراءات مختلفة)، اما الحقل مصدر البيانات dataSrc[5] فهو مصفوفة عناصرها من النوع unsigned char وتحتوي 5 مواقع (تسمح بتخزين عناوين 5 عقد)، ان كل موقع من المصفوفة مصدر البيانات يحوي عنوان العقدة التي توجد قراءتها في الموقع الموافق من مصفوفة البيانات. أي أن القراءة الموجودة في الموقع data[i] تم ارسالها من قبل العقدة الموجود عنوانها في الموقع dataSrc[i].

كما قمنا ببرمجة بروتوكول SAR بأبسط أشكاله وهو عبارة عن بروتوكول Proactive يصنف على أنه بروتوكول زمن حقيقي متعدد المسارات. كما قمنا بتعريف البروتوكول بحيث تقوم العقدة بشكل تلقائي بتحديد العقدة التي من خلالها يمكن الوصول الى المصرف (Sink) كما أن العقد التي تنضم حديثا الى الشبكة تستطيع أيضا تحديد الوجهة للوصول الى المصرف وبأقل عدد قفزات. قمنا باستخدام متغيرات لتخزين معلومات التوجيه في كل عقدة وفق بروتوكول SAR كما يظهر الجدول التالي:

جدول 2: المتحولات الخاصة بتخزين معلومات التوجيه في العقد

الوصف	حجم المتحول	نوع المتحول	اسم المتحول
عدد القفزات للوصول الى المصرف	1 Byte	Unsigned char	hopCount
عنوان العقدة	1 Byte	Unsigned char	address
العقدة التالية (العقدة الاب)	1 Byte	Unsigned char	parent

في بداية عمل الشبكة تكون قيمة المتحول parent لكل عقد الشبكة هو 255 أي أن العقدة لا تملك عقدة أب وبالتالي ليس لديها اتصال مع عقدة المصرف، وبحسب خوارزمية البروتوكول لا يتم ارسال أي قراءة في هذه الحالة بل تبقى العقد في حالة استماع. يقوم المصرف بإرسال رسالة تهيئة بحقل عدد قفزات يساوي الصفر (hopCount=0) بطريقة البث العام تستقبلها العقد المحيطة بالمصرف وتقوم بتخزين عنوان المصرف في المتغير parent، وتحتفظ عدد القفزات hopCount=1. تقوم بعدها العقد السابقة (المحيطة بالمصرف) و وفق جدول ارسال القراءات تقوم بإرسال القراءة و يتم

تحديد حقول الرزمة بحيث يكون عدد القفزات هو $hopCount=1$ و عنوان الهدف هو عنوان المصرف, تقوم العقد المحيطة باستقبال هذه الرزمة على الرغم من أنها غير موجهة لها , و تقوم بمقارنة حقل ال $hopCount$ المخزن فيها مع حقل ال $hopCount$ في الرزمة المستقبلية بعد زيادته بمقدار واحد , فإذا كانت قيمة الحقل الذي في الرزمة أقل من القيمة المخزنة في العقدة يتم تخزين عنوان العقدة المرسل كعنوان أب (Parent) و يتم تحديث قيمة عدد القفزات. تتكرر هذه العملية حتى تصبح كل عقد الشبكة لديها عقدة أب ترسل من خلالها الى المصرف وبأقل عدد قفزات ممكن.

1-1-3 تعامل المحاكى مع برنامج خارجي في الزمن الحقيقي من خلال الصنف :SocketRTScheduler

للتعامل مع محاكي ال OMNET++ من خارجه و في الزمن الحقيقي نحتاج لقناة تربط المحاكى مع هذا البرنامج , في هذا المشروع كان هدف الربط هو نقل الرزم من المحاكى باتجاه دارة الربط لايصالها لاحقاً للعقد الحقيقية , وكذلك اىصال الرزم القادمة من العقد الحقيقية عبر دارة الربط الى المحاكى , و يجب أن تتم هذه العملية في الزمن الحقيقي لتحقيق الهدف من عملية المحاكاة .

استخدمنا في هذا المشروع الصنف SocketRTScheduler وهو مأخوذ من أحد الأمثلة المرفقة في برنامج ال OMNET++. ان هذا الصنف تمت وراثته من الصنف cScheduler و هو الصنف المسؤول عن عملية تنظيم الأحداث أثناء المحاكاة , و في المحاكى OMNET++ و بشكل افتراضي عملية تنظيم الاحداث تتم بشكل تسلسلي حيث يأخذ المتغير scheduler-class ضمن ملف اعدادات المشروع omnet.ini و بشكل افتراضي القيمة cSequentialScheduler .

بتغيير قيمة هذا المتحول الى "cSocketRTScheduler" scheduler-class = , تصبح عملية تنظيم الاحداث مرتبطة بالزمن الحقيقي كما أن العقدة في الشبكة و التي تستخدم هذا الصنف في تابع التهيئة لها تتحول الى مخدم TCP يعمل على المنفذ المحدد بقيمة المتغير RTS_PORT ضمن ملف ال GLOBALS.h , ويتم استخدام هذا الصنف في تابع التهيئة للعقدة كما يلي :

```
void MyInterfaceModule::initialize()
{
    extEvent = new cMessage("extEvent");
    rtScheduler = check_and_cast<cSocketRTScheduler *>(simulation.getScheduler());
    rtScheduler->setInterfaceModule(this, extEvent, buf, 4000, numBytes);
}
```

وذلك بعد تعريف المتغيرات التالية في الصنف الخاص بالعقدة:

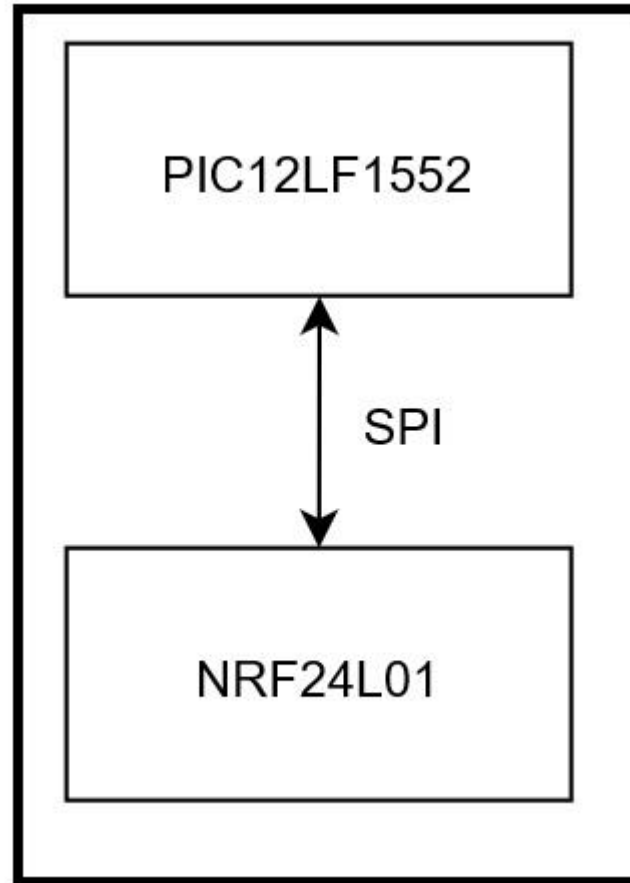
```
cSocketRTScheduler *rtScheduler;
cMessage *extEvent;
char buf[4000];
int numBytes;
```

عند ورود معلومات من البرنامج المتصل بهذا المخدم , يتم ارسال حدث داخلي في العقدة باسم "extEvent" ويتم التعامل معه في التابع handleMessage كأى حدث اخر , وللحصول على المعلومات يجب القراءة من المتغير .buf

2-1 تصميم العقد الحقيقية

العقدة الحقيقية هي عقد الحساسات اللاسلكية الموجودة ضمن البيئة الحقيقية وهي مؤلفة من جهاز الارسال والاستقبال اللاسلكي NRF24L01 والمتحكم PIC12LF1552 الذي تم اختياره بناءً على متطلبات الاستهلاك المنخفض للطاقة. يستخدم المتحكم السابق من اجل إدارة جهاز الارسال والاستقبال اللاسلكي ويتم الاتصال بين المتحكم وجهاز الارسال والاستقبال اللاسلكي عبر بروتوكول الاتصال التسلسلي بالطرفيات (SPI (Serial Peripheral Interface).

الشكل التالي يمثل مخطط صندوقي لعقدة حقيقية:



رسم توضيحي 1: مخطط صندوقي لعقدة حقيقية

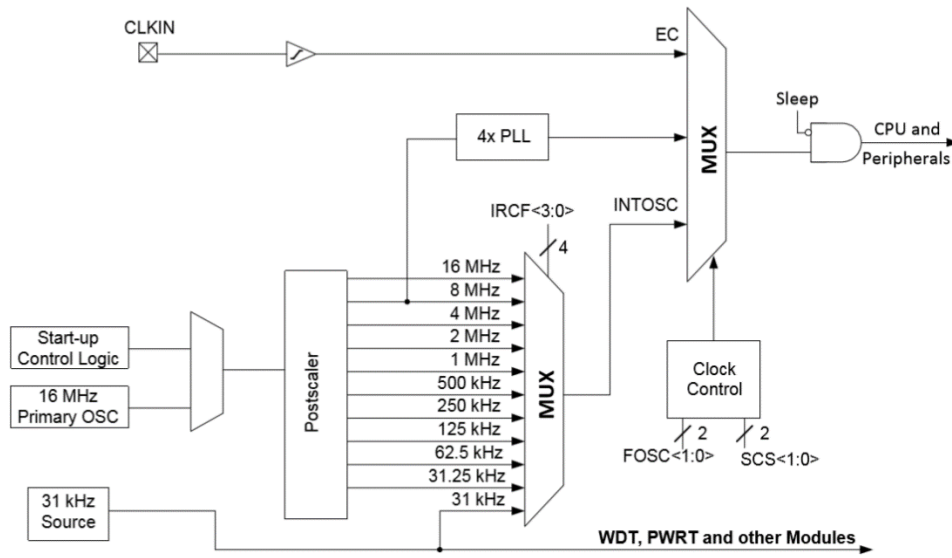
1-2-1 المتحكم PIC12LF1552

وهو المتحكم المستخدم في الوحدات اللاسلكية وذلك لإدارة عمل وحدة الارسال/استقبال NRF24L01، من أهم

مميزاته:

- توفير الطاقة، حيث يستهلك 30 uA/MHz عند جهد 1.8V وذلك في وضعية العمل الطبيعي.

- هزاز داخلي بتردد 16 MHz، وبوجود حلقة اقفال طوري (Phase-Locked Loop) يمكن اختيار تردد العمل من 31 KHz حتى 32 MHz، أما في حالة العمل بوضع توفير الطاقة فيمكن اختيار تردد هزاز داخلي 31 KHz.
 - إمكانية وصل هزاز خارجي، حيث يعمل المتحكم عندها بأحد الانماط التالية.
 - وضع استهلاك الطاقة المنخفض: التردد الخارجي من 0 حتى 0.5 MHz.
 - وضع استهلاك الطاقة المتوسط: التردد الخارجي من 0.5 حتى 4 MHz.
 - وضع استهلاك الطاقة العالي: التردد الخارجي من 4 حتى 20 MHz.
- ويظهر الشكل التالي وحدة الهزاز داخل المتحكم:



رسم توضيحي 2: وحدة الهزاز داخل المتحكم PIC12LF1552

- مجال التغذية كبير وهو من 1.8V حتى 3.6V.
- يحوي المتحكم محول تشابهي رقمي (ADC) يعطي قيم بدقة 10 bit ويدعم 5 قنوات خارجية وقناتين داخليتين، إحدى القناتين الداخليتين تتصل مع مقياس لدرجة الحرارة، وهو التطبيق الذي تم استعماله في المشروع.
- يدعم واجهة الربط التسلسلية مع المحيطيات (Serial peripheral interface) والمستخدم للاتصال بالوحدة .NRF24L01

3-1 دارة الربط بين البيئة الحقيقية والبيئة الافتراضية

وتدعى أيضا العقدة الهجينة لأنها مؤلفة من قسمين: حقيقي وافتراضي (برمجي)، القسم الحقيقي فيتألف من دارة تحوي المتحكم PIC16F877A وجهاز الارسال والاستقبال NRF24L01 والدارة المتكاملة MAX232 بالإضافة الى العناصر غير الفعالة اللازمة لعمل الأجزاء السابقة. يستخدم جهاز الارسال والاستقبال NRF24L01 من اجل الاتصال مع بقية العقد الحقيقية في الشبكة، ويتم التحكم به عن طريق المتحكم PIC16F877A باستخدام بروتوكول الاتصال SPI،

أما المتحكم فيتم ربطه مع الحاسب (الذي يعمل عليه المحاكى) باستخدام بروتوكول الاتصال غير المتزامن UART وبسبب أن الوصل تم باستخدام الوصلة RS232 التي تعمل عند المستويات المنطقية +13 و -13 لتمثيل كل من الصفر والواحد المنطقي على التوالي وهي تختلف عن المستويات التي يعمل عندها المتحكم (+5v و 0v) لتمثيل كل من الواحد والصفر المنطقي، كانت الحاجة إلى الدارة المتكاملة MAX232 التي تقوم بالتحويل بين المستويات المنطقية لهذه الدارات المختلفة. أما القسم الافتراضي لهذه العقدة فيتألف من قسمين: القسم الأول يظهر في المحاكى على شكل عقدة هي عبارة عن وحدة بسيطة، عند استدعاء التابع initialize لهذه العقدة يتم إنشاء مخدم (server) يعمل على العنوان المحلي برقم منفذ معين. أما القسم الثاني فهو عبارة عن برنامج عميل (client) يعمل خارج المحاكى (تمت كتابته بلغة Python) ويتصل بالمخدم السابق كما يتصل أيضا بالمتحكم PIC16F877A (عبر UART).

الفصل الثاني:

الواجهات البرمجية للبيئة الافتراضية

تم تقسيم الكود الخاص بالعقد الافتراضية التي تعمل ضمن المحاكى الى مكتبات تحوي توابع بحسب الطبقة التي تعمل فيها هذه التوابع وبالتالي كان لدينا الطبقتان التاليتان:

1-2 طبقة الشبكة Network Layer

هذه الطبقة تضم بروتوكول التوجيه (البروتوكول المستخدم هو Sequential Assignment Routing SAR) حيث أن كل عقدة يكون لها عنوان مميز address يتم إعطائه للعقدة عند انشاء الشبكة، والبارامتر المستخدم لتحديد الطريق الأفضل هو عدد القفزات hopCount بحيث عندما تصل الى العقدة رسائل من جيرانها (ليس بالضرورة أن تكون مرسلة إلى تلك العقدة) تقوم بحساب بارامتر عدد القفزات الجديد (تكون قيمته البدائية 255 أي لا نهاية) في حال كان عدد القفزات الجديد أقل من عدد القفزات القديم تقوم تلك العقدة بتعيين العقدة التي تلقت منها الرسالة كعقدة أب parent وتقوم بإرسال رسائلها الى الـ sink عبر الـ parent الخاص بها. وفي حال اكتشافها لعقدة أب تمتلك عدد قفزات أقل تقوم بتعيين هذه العقدة الجديدة كأب لها وتتم عملية الارسال عن طريق هذه العقدة الجديدة. يظهر الجدول التالي المتغيرات التي تم تعريفها في طبقة الشبكة:

جدول 3: المتغيرات الخاصة بطبقة الشبكة

الوصف	حجم المتحول	نوع المتحول	اسم المتحول
عدد القفزات للوصول الى المصرف	1 Byte	Unsigned char	hopCount
عنوان العقدة	1 Byte	Unsigned char	address
العقدة التالية (العقدة الاب) للوصول الى المصرف	1 Byte	Unsigned char	Parent

يوجد بشكل أساسي ثلاثة توابع تعمل ضمن هذه الطبقة يتم استدعاء كل منها بحسب الاحداث الواردة الى هذه العقدة وهي:

- a- التابع sendNewPacket يقوم بإنشاء رزمة جديدة ويملاً حقولها كما يلي:
 - عنوان المرسل = address (عنوان العقدة التي تم فيها استدعاء التابع sendNewPacket)
 - عنوان الهدف = parent
 - عدد القفزات = hopCount
 - عدد القيم (الرسائل ضمن هذه الرزمة) = 1 (لأنها رزمة جديدة لا تحوي أي رسائل سابقة)
 - كما يقوم بوضع القراءة المأخوذة من قبل العقدة في الموقع الأول من مصفوفة البيانات وعنوان هذه العقدة في الموقع الأول من مصفوفة مصدر البيانات.

b- التابع `handleMyPacket` يتم استدعاؤه عند ورود رزمة جديدة مرسلة الى العقدة (أي حقل `dstAddress` في الرزمة المستقبلية يساوي عنوان العقدة `address` التي استقبلت الرسالة) لتقوم بعدها العقدة بالتحقق من عدد الرسائل الموجود ضمن هذه الرزمة فإن كان أصغر من 5 تضيف العقدة قراءتها وعنوانها الى الرزمة وتقوم بتمريرها الى عقدة الاب الخاصة بها، اما في حال كان عدد العقد يساوي 5 فيتم تمرير الرزمة المستقبلية كما هي الى عقدة الاب ويتم استدعاء التابع `sendNewPacket`.

c- التابع `handleOverHeard` يتم استدعاء عند استقبال رزمة لا تخص العقدة التي استقبلتها (أي ان حقل `dstAddress` في الرزمة المستقبلية لا يساوي عنوان العقدة `address` التي استقبلت تلك الرزمة) ليقوم بمقارنة عدد القفزات المسجل ضمن العقدة مع قيمة حقل عدد القفزات للرزمة المستقبلية بعد زيادته بمقدار واحد، فإن كان عدد القفزات للرزمة اقل من عدد القفزات المسجل ضمن العقدة يتم تعيين العقدة التي أرسلت الرزمة كعقدة أب. الجدول التالي يوضح هذه التوابع باختصار:

جدول 4: التوابع الخاصة بطبقة الشبكة

الوصف	البارامترات	اسم التابع
تابع ارسال رزمة جديدة بمعلومات هي البارامتر <code>data</code>	<code>Float data</code>	<code>sendNewPacket</code>
تابع لمعالجة الرزمة المستقبلية <code>pkt</code> والتي تم ارسالها من قبل عقدة اخرى بشكل صريح للعقدة الحالية	<code>Packet *pkt</code>	<code>handleMyPacket</code>
تابع لمعالجة الرزمة المستقبلية <code>pkt</code> والتي استقبلتها العقدة دون ان تكون مرسلة اليها	<code>Packet *pkt</code>	<code>handleOverHeard</code>

2-2 طبقة التطبيقات Application Layer

في هذه الطبقة يتم تنفيذ جزء البرنامج الخاص بعملية أخذ العينات في حالة العقد الحقيقية سيتم قراءة العينات عن طريق دخل المحول التشابهي الى رقمي أما في حال كون العقدة افتراضية في المحاكي فيتم توليد رقم عشوائي في المجال 0 الى 70 يمثل قراءة الحساس عند هذه العقدة. وهذه الطبقة تحوي التابعين التاليين:

a- التابع `generateData`: يقوم بمحاكاة عملية أخذ قراءة حساس، ويقتصر عمله في الواقع على توليد قيمة من نوع `float` تقع في المجال من 0 الى 70 وهي تحاكي درجة الحرارة عند العقد الافتراضية.

b- التابع `scheduleNextEvent` يقوم بعمل جدول (أي تحديد موعد) لعملية اخذ قراءة الحساس التالية أي يحدد الفاصل الزمني الواجب انقضاءه قبل الاستدعاء التالي للتابع `generateDate`.

الفصل الثالث:

الواجهات البرمجية للبيئة الحقيقية

تتألف العقدة الحقيقية من جهاز ارسال واستقبال NRF24L01 يتم التحكم به عبر متحكم PIC1552 وذلك عبر بروتوكول الاتصال SPI. قمنا بتقسيم الكود البرمجي للمتحكم الى مكتبات تمثل الطبقات التي يمكن الوصول اليها والتحكم بها في جهاز الارسال والاستقبال هي الطبقة الفيزيائية وطبقة نقل المعطيات وطبقة الشبكة وطبقة التطبيقات، حيث كل طبقة تحوي التوابع الملائمة لهذه الطبقة. الشكل التالي يوضح هذه المكتبات:

Network
Datalink
Physical Layer
NRF_Tools
NRF24L01

رسم توضيحي 3: المكتبات التي يتألف منها كود المتحكم PIC1552

3-1 مكتبة NRF24L01

تحتوي هذه المكتبة مجموعة الثوابت التي تستخدم للتحكم بالنRF24L01، هذه الثوابت تشمل تعليمات التحكم وعناوين المسجلات ضمن الNRF24L01 بالإضافة الى الخانات ضمن هذه المسجلات. مثلا التعليمة W_REGISTER تأخذ الشكل: 001AAAAA حيث يتم استبدال AAAAAA برقم المسجل المراد الكتابة عليه أو القراءة منه، مثلا للكتابة على المسجل STATUS، ذو العنوان 0x07 يتم ارسال الامر W_REGISTER + STATUS حيث إشارة ال+ تمثل عملية OR وقيمة الثابت W_REGISTER هي 0x01، أما قيمة الثابت STATUS هي 0x07 بالتالي فإن الأمر المرسل هو 0x17.

3-2 مكتبة NRF_Tools

تحتوي هذه المكتبة على توابع معرفة للتحكم بالنRF24L01، الجدول التالي يظهر هذه التوابع:

جدول 5: التوابع ضمن المكتبة NRF_Tools

اسم التابع	البارامترات	وصف التابع
toggle	لا يوجد بارامترات	قذح الرجل Chip Enable لبدء عملية الارسال
toggle_CSN	لا يوجد بارامترات	قذح الرجل CSN من أجل الكتابة على مسجلات الNRF24L01
get_STATUS	لا يوجد بارامترات	يعيد قيمة المسجل STATUS
get_FIFO_STATUS	لا يوجد بارامترات	يعيد قيمة المسجل FIFO_STATUS

get_CD	لا يوجد بارامترات	لكشف إذا كان هناك عملية ارسال من عقدة مجاورة (قراءة المسجل (Carrier Detect
NRF_RW	unsigned char command unsigned char dat	لتنفيذ الأوامر المتاحة في ال NRF ، حيث البارامتر command هو الامر الذي سيتم تنفيذه، و dat هو المعلومات التي يتم تمريرها، الاوامر المتاحة موجودة ضمن النشرة الفنية لجهاز ال NRF .
TX_mode	لا يوجد بارامترات	وضع ال NRF24L01 في وضع الارسال
RX_mode	لا يوجد بارامترات	وضع ال NRF24L01 في وضع الاستقبال

3-3 الطبقة الفيزيائية Physical layer

تختص هذه الطبقة بتحديد معدل نقل البيانات في جهاز ال NRF24L01 وتحديد تردد القناة العاملة وتحديد ربح الاستقبال (حساسية المستقبل) وتحديد استطاعة الارسال.

يظهر الجدول التالي التوابيع المعرفة ضمن هذه الطبقة:

جدول 6: توابيع الطبقة الفيزيائية للبيئة الحقيقية

وصف التابع	البارامترات	اسم التابع
يتم تحديد معدل نقل البيانات من خلال البارامتر rate والذي يأخذ القيم 2 من أجل معدل 2Mbps و 1 من أجل معدل 1Mbps .	Unsigned char rate	Set_data_rate
يتم تحديد تردد القناة العاملة من خلال البارامتر ch والذي يأخذ القيم 0 حتى 125 بحيث يكون تردد القناة وفق المعادلة $2400 + ch$ MHz	Unsigned char ch	Set_RF_CH
يتم تحديد ربح المستقبل من خلال البارامتر gain والذي يأخذ القيم 0 من	Unsigned char rate	RX_LNA_gain

		أجل ربح منخفض (استهلاك منخفض للطاقة) و 1 من أجل ربح عال.
TX_output_power_control	Unsigned char pwr	يتم تحديد استطاعة المرسل من خلال البارامتر pwr والذي يأخذ القيم 0 حتى 3، حيث أن القيمة 0 للاستطاعة المنخفضة.

3-4 طبقة نقل المعطيات DataLink Layer

تحتوي هذه الطبقة التوابع الخاصة بعمليات تشكيل الرزمة من تحديد إذا ما كانت حمولة الرزمة من المعطيات ثابتة الطول أو ديناميكية، وتحديد عرض العناوين المستخدمة في هذه الطبقة (والتي تختلف عن عناوين العقد في طبقة الشبكة)، وتفعيل أو إلغاء تفعيل الاخطار التلقائي بوصول الحزمة (Auto Acknowledgment)، وتحديد عدد البايتات المستخدمة في توليد كود التحقق من الحزمة Cyclic redundancy check (CRC). باستخدام تلك التوابع قمنا ببرمجة طبقة فرعية لهذه الطبقة تشكل واجهة بسيطة لها تحوي التابعين transmit و recv وذلك وفق إعدادات محددة وهي:

- إلغاء الاخطار التلقائي.
- استخدام عنوان البث الافتراضي.
- استخدام طول معطيات ثابت وهو الطول الاعظمي 32Byte.
- طول كود ال CRC هو 1Byte.

يظهر الجدول التالي ملخص للتابعين transmit و recv:

جدول 7 توابع طبقة نقل المعطيات للبيئة الحقيقية

الوصف	البارامترات	اسم التابع
تابع الارسال، يقوم بإرسال محتوى المصفوفة buffer ذات الطول PW.	Unsigned char *buffer , unsigned char _PW	Transmit
تابع الاستقبال، يقوم بتخزين المعطيات ذات الطول PW في المصفوفة buffer	Unsigned char *buffer , unsigned char _PW	recv

3-5 طبقة الشبكة Network layer

يتم في هذه الطبقة تعريف التوابع التي تخص بروتوكول التوجيه، والتوابع المستخدمة هنا تطابق التوابع المستخدمة في المحاكى وذلك لتسهيل عملية التعديل على بروتوكول التوجيه أو تغييره، والتوابع الخاصة بهذه الطبقة موضحة في الجدول التالي:

جدول 8: توابع طبقة الشبكة في البيئة الحقيقية

الوصف	البارامترات	اسم التابع
-------	-------------	------------

sendNewPacket	لا يوجد بارامترات	تابع ارسال رزمة جديدة بمعلومات هي المتحول ApplicationData من طبقة التطبيقات
handleMyPacket	لا يوجد بارامترات	تابع لمعالجة الرزمة والتي تخزن في المتحول dat والذي يشكل عازل Buffer لتخزين المعلومات في حالتي الارسال والاستقبال
handleOverHeard	لا يوجد بارامترات	تابع لمعالجة الرزمة المستقبلية والمخزنة في المتحول dat والتي استقبلتها العقدة دون ان تكون مرسلة اليها

3-6 طبقة التطبيقات

وفي هذه الطبقة يتم استخدام توابع لقراءة الحساس ومعالجتها في المتحكم، وفي مشروعنا قمنا باستخدام حساس الحرارة الداخلي للمتحكم واستخدمنا العداد TIMER0 ليقوم المتحكم بإرسال القراءة كل 10 ثواني.

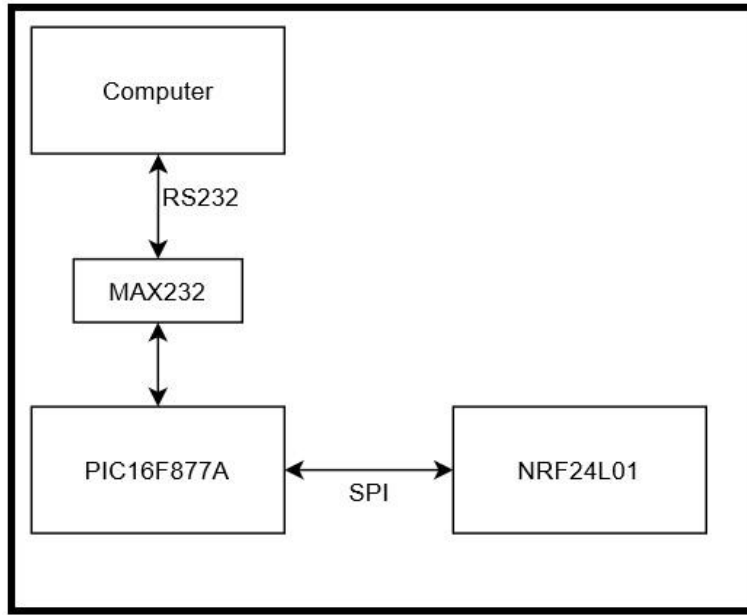
الفصل الرابع:

دارة الربط بين البيئتين الافتراضية والحقيقية

كما ذكرنا سابقا تسمى هذه الدارة بالعقدة الهجينة لأنها مؤلفة من قسمين حقيقي وبرمجي (افتراضي)، وتتكون بشكل أساسي من المتحكم PIC16F877A وجهاز الارسال والاستقبال NRF24L01 ودارة الملائمة MAX232.

1-4 عناصر الدارة

الشكل التالي يوضح هذه الدارة:



رسم توضيحي 4: مخطط صندوقي لدارة الربط

1-1-4 المتحكم PIC16F877A

تم اختيار هذا المتحكم لأنه يدعم بروتوكول الوصل التسلسلي SPI من أجل الربط مع NRF24L01 وبروتوكول الاتصال اللاسلكي UART من أجل الربط مع الحاسب. قمنا بتشغيل هذا المتحكم مستخدمين هزاز خارجي على التردد 8MHz. واستخدمناه للتواصل مع الـ NRF24L01 وإدارتها عن طريق الحاسب.

يحتوي برنامج المتحكم على توابيع للتحكم بجهاز الارسال والاستقبال، وقمنا ببرمجته ليتلقى أوامر من برنامج يعمل على الحاسب لتنفيذ أحد تلك التوابيع وقراءة النتائج المعادة من جهاز الارسال والاستقبال وإعادة إرسالها الى الحاسب (من أجل عملية كشف الأخطاء).

1-2-4 جهاز الارسال والاستقبال NRF24L01

وهي وحدة الارسال/استقبال المستخدمة في العقد اللاسلكية في المشروع، من أهم ميزاتها:

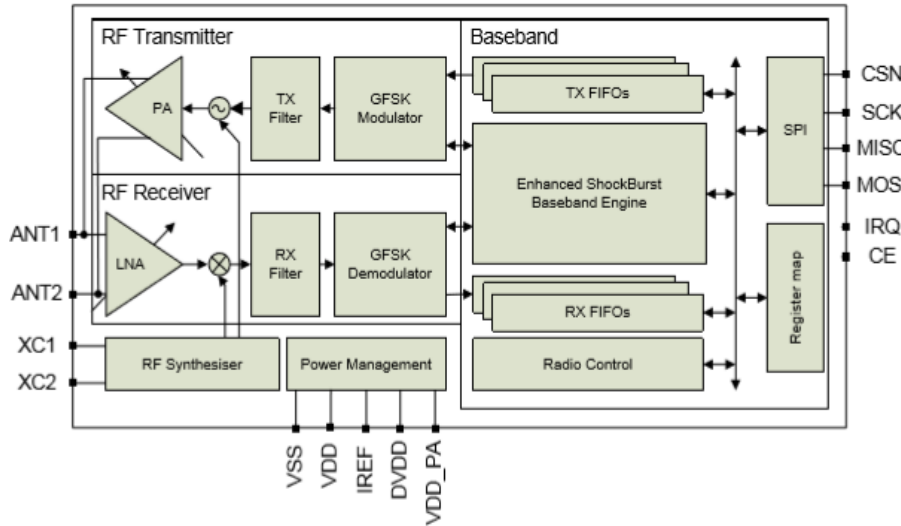
- أنها تعمل على الحزمة المخصصة للتطبيقات الطبية والعلمية والصناعية (ISM Band) وذلك على التردد 2.4GHz.

- كما أنها تقدم معدل نقل بيانات حتى 2Mbps.
- وتعتبر وحدة موفرة للطاقة حيث تستهلك 900nA في حالة الخمول (Power Down) و 22uA في وضع الاستعداد.
- مجال تغذية من 1.9V حتى 3.6V.
- الهزاز العامل في الوحدة هو 16Mhz.
- مستويات الجهد كدخل من الدارات الأخرى يبلغ 5V.
- تحوي طبقة Data Link مما يسهل التعامل معها ويقلل العبء على المتحكم الخارجي.
- تعمل على بروتوكول نقل خاص هو Enhanced ShockBurst الذي يتيح حقلا مخصصا للحمولة ضمن الرزمة المرسله بطول يتراوح بين 1 و 32 بت، كما يدعم الإرسال الموثوق عن طريق الاخطار باستلام الرسالة.
- تم تصميم الوحدة nrf24l01 لكي تعمل في مجال ترددات الISM ضمن التطبيقات اللاسلكية ذات الطاقة المنخفضة، وتحتاج لمتحكم خارجي ويضع عناصر أخرى فقط لتحقيق النظام اللاسلكي. يتم ضبط والتعامل مع الوحدة nrf24l01 من خلال واجهة الطرفيات التسلسلية (Serial Peripheral Interface)، حيث تسمح هذه الواجهة بالدخول الى خريطة المسجلات (Register Map) الخاصة بالوحدة والتعديل عليها، ويتم ذلك من خلال أربعة أطراف وبمعدل نقل لأعظمي 8Mbps.

التعديل المستخدم في الوحدة عند الإرسال الراديوي هو GFSK، والذي يمكن ضبط بارامتراته من تردد القناة واستطاعة الإرسال ومعدل النقل. ان عدد الاقنية المتاحة هو 126 قناة يبدأ تردد الاقنية من 2400Mhz حتى 2525Mhz، ومعدلات نقل المعطيات التي يمكن اختيارها هي 2Mbps (وهو المعدل الافتراضي) و 1Mbps.

في وضع الإرسال يمكن اختيار قدرة الخرج من الخيارات التالية 0، -6، -12، -18 dBm . حيث يبلغ سحب التيار في حالة 0dbm يبلغ 11.3mA، أما في وضع الاستقبال يبلغ سحب التيار 12.3mA عند اختيار معدل نقل بيانات 2Mbps.

الشكل التالي يوضح مخطط صندوقي للـ NRF24L01 :



رسم توضيحي 5: المخطط الصندوقي للـ NRF24L01

امن أجل عمل هذه الوحدة كمرسل مستقبل يجب ان يتم وضعها في دارة مؤلفة من بضعة عناصر غير فعالة بالإضافة الى متحكم كما يجب تأمين هوائي يستخدم للإرسال والاستقبال ومن أجل صغر الحجم ومتطلبات الطاقة

المنخفضة يستعمل هوائي الدارة المطبوعة والذي هو عبارة عن مسار مرسوم على اللوحة المطبوعة، وله عدة اشكال فقد يكون خط مستقيم، او بشكل الحرف F معكوس، أو بشكل دائري، او منحني.

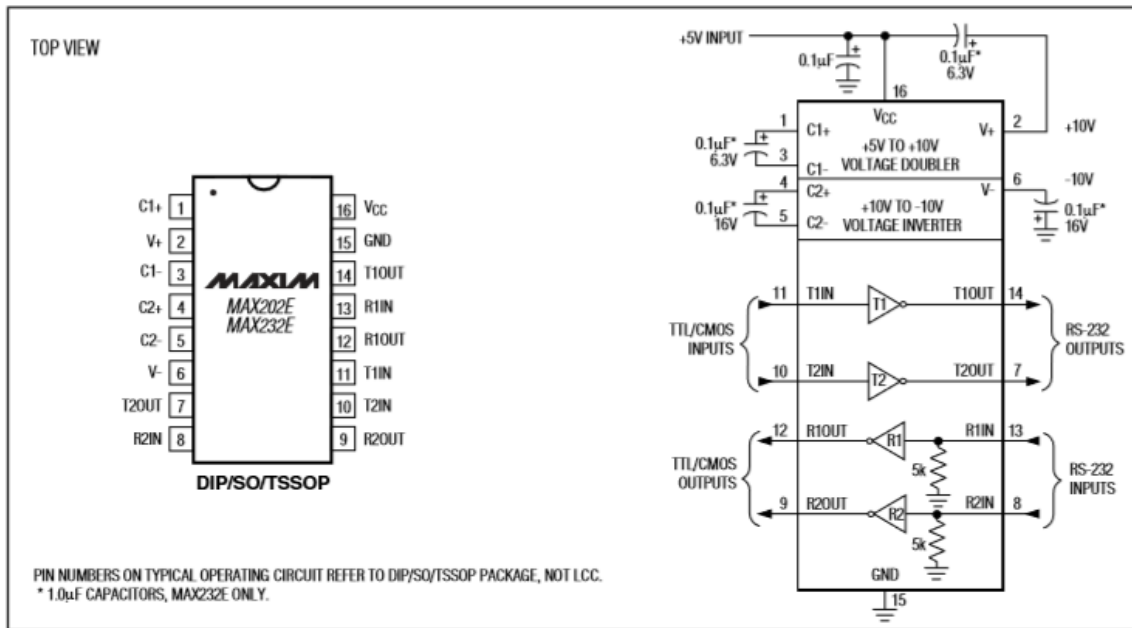
يعتبر هوائي الدارة المطبوعة هوائيا ثنائي البعد، كما انه اقل تكلفة من الهوائي السلكي لكن في المقابل فانه اقل كفاءة ويعمل عند استطاعات منخفضة.



رسم توضيحي 6: هوائي الدارة المطبوعة

3-1-4 دائرة الملائمة MAX232

تعمل هذه الدارة على تحقيقي الملائمة بين الجهود التي يعمل عليها المتحكم وهي +5v لتمثيل الواحد المنطقي 0v ولتمثيل الصفر المنطقي والجهود الخاصة ببروتوكول RS232 وهي +13v لتمثيل الصفر المنطقي و -13v لتمثيل الواحد المنطقي. الشكل التالي يوضح مخطط هذه الدارة:



رسم توضيحي 7: دائرة الملائمة MAX232

2-4 الواجهة البرمجية لدائرة الربط

يتم التحكم بالمتحكم الصغري PIC16F877A عن طريق الحاسب ببرنامج مكتوب بلغة Python بعد أن يتم وصل المتحكم إلى الكمبيوتر عبر وصلة RS232. حيث أن برنامج المتحكم يتضمن التتابع المطلوبة لعمل المتحكم مع جهاز الارسال والاستقبال وعند تشغيل المتحكم سيدخل قريبا في حلقة while وينتظر الأوامر من الحاسب، هذه الأوامر هي عبارة عن

محارف Characters كل محرف يوافق تنفيذ تابع من التوابع المبرمجة على المتحكم (مع ارسال بارامترات التابع في حال كان ذلك التابع يتطلب بارامترات).

المثال التالي للتوضيح فقط وهو عبارة عن جزء صغير من الكود المستخدم في المتحكم PIC16F877A:

```
while (1) {
    command=my_UART1_Read();
    switch (command) {
        case '5': toggle ();
            break;
        case 'e':arg1=my_UART1_Read ();
            set_data_rate (arg1);
            break;
    }
}
```

التابع my_UART1_Read يقوم بالقراءة من عازل Buffer الـ RS232 واستخلاص القيمة المرسله بعد ذلك يتم تخزينها في المتحول command ويتم ارسالها إلى تعليمة switch التي تنفذ التابع الموافق للقيمة المرسله، فمثلا في حال أردنا تنفيذ التابع toggle يتم ارسال المحرف 5 من الكمبيوتر المتصل، وفي حال كان التابع يحتاج إلى بارامترات يتم ارسال البارامترات المطلوبة بعد ارسال المحرف الموافق لتابع المراد تنفيذه.

بالعودة إلى المثال السابق ومثلا لتنفيذ التابع set_data_rate الذي يقوم بتحديد معدل ارسال البيانات في الـ NRF24L01 والذي يحتاج بارامترات واحدا هو معدل البيانات المراد ضبطه، يجب أن يرسل المحرف e وعندما يقرأ المتحكم المحرف e سينفذ التعليمة arg1=my_UART1_Read حيث سينتظر ارسال قيمة البارامتر arg1 من الحاسب، وبعد وصول القيمة سيتم تخزينها في المتحول arg1 والانتقال لتنفيذ التعليمة set_data_rate(arg1).

بالنسبة لبرنامج الحاسب ومن أجل التواصل مع المتحكم عن طريق وصلة الـ RS232 يتم استخدام المكتبة pyserial في البرنامج ، ويتم الاتصال بدارة الربط بعد تحديد معدل البود و رقم المنفذ:

```
serial_com=serial.Serial('com1', 128000)
```

في البرنامج المستخدم قمنا بإنشاء صنف Class يدعى nrf24l01_pc_base يضم عدة توابع منها تابع exe والذي يقوم بإرسال الأوامر إلى المتحكم في دارة الربط ويستقبل النتيجة المعادة (في حال وجود قيمة معادة).

ومن التوابع الخاصة بالصنف nrf24l01_pc_base التابعين recv من أجل قراءة القيمة المستقبلية و transmit من أجل كتابة القيمة المراد إرسالها وهذان التابعين الاخيران يعتمدان في عملهما على التابع exe.

بالعودة للتابع exe وباعتباره التابع الأساسي في عملية إدارة عمل المتحكم PIC16F877A في دارة الربط قمنا بتعريف التابع على الشكل التالي:

```
exe(self, command, args, have_result=0)
```

حيث أن command يتم استبدالها باسم الأمر المراد تنفيذه وباعتبار أن تذكر جميع المحارف المستخدمة والتتابع الموافقة لكل محرف ليس امرا سهلا تم تعريف عدة ثوابت يكون اسمها ذو دلالة على التابع الموافق لها وتحتوي قيمة المحرف الموافق لذلك التابع. مثلا في حال الرغبة بتنفيذ التابع set_data_rate في المثال السابق وبدلا من ارسال المحرف e تم التعريف الثابت

```
SET_DATA_RATE='e'
```

بالنسبة لـ args فهي عبارة عن مصفوفات البارامترات اللازمة لعمل التابع المراد تنفيذه أما البارامتر الأخير have_result فيستخدم لتحديد ما إذا كان التابع الذي تم تنفيذه يعيد ارسال قيمة إلى الحاسب أم لا حيث أن قيمته الافتراضية هي 0 اي ان التابع المنفذ لا يرسل أي قيمة وفي حال كان التابع المنفذ سيقوم بإرسال قيمة يجب اعطاء هذا البارامتر القيمة 1.

مثلا لتنفيذ التابع set_data_rate يمكن كتابة التعليمة التالية في محرر البايثون:

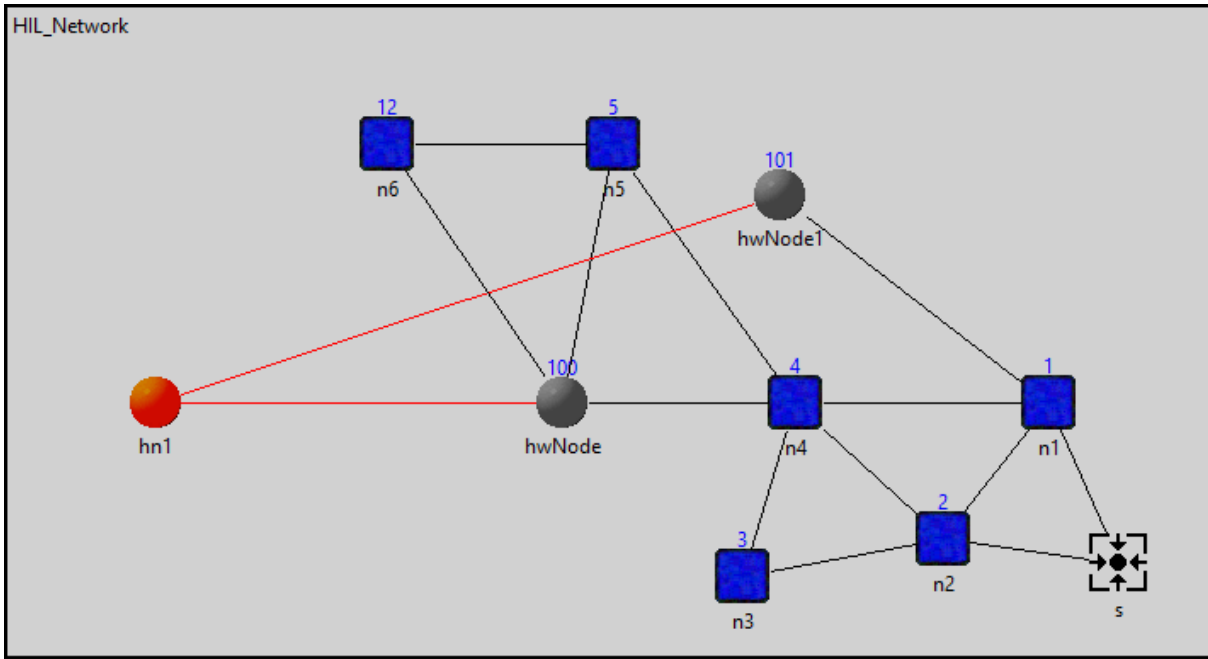
```
NrfObject.exe(SET_DATA_RATE, [1])
```

حيث أن SET_DATA_RATE='e' والبارامتر المرسل هو 1 وتنفيذ التابع set_data_rate لا يعيد أي قيمة.

الفصل الخامس: مخططات الشبكة

1-5 مخطط الشبكة في المحاكى

المخطط التالي يمثل شبكة تكون فيها العقد الزرقاء عقدا افتراضية والعقد الرمادية هي العقد الحقيقية والعقدة الحمراء هي العقدة الهجينة أما العقدة التي تبدو على شكل دائرة تشير إليها أربعة أسهم فهي عقدة الـ Sink.



رسم توضيحي 8: مخطط شبكة الحساسات اللاسلكية المدروسة

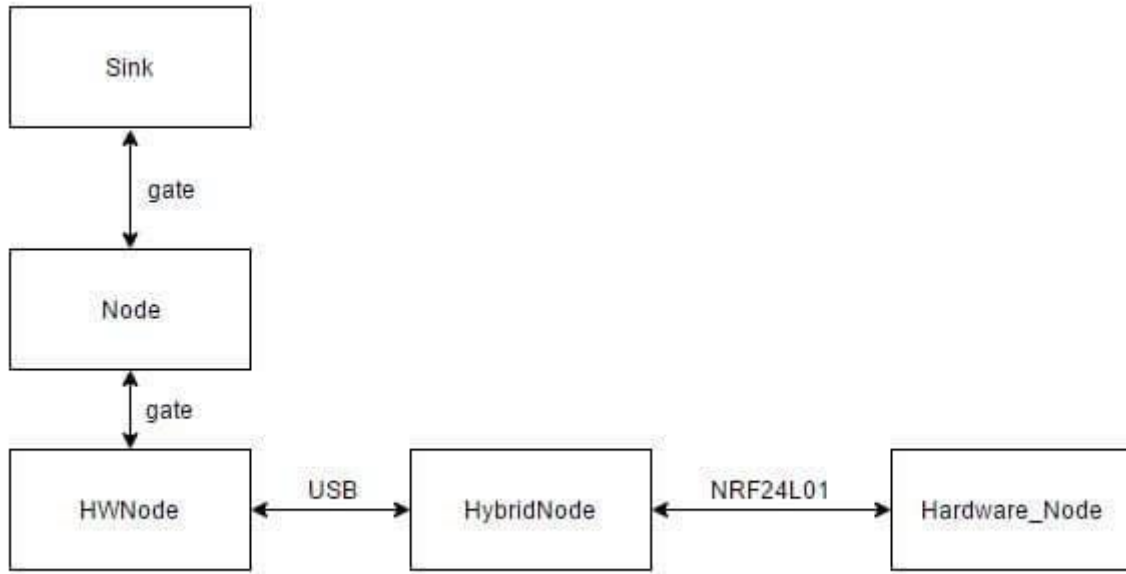
كما يظهر في الشكل السابق فإن العقد باللون الأزرق هي العقد الافتراضية (التي تم محاكاتها) والعقد باللون الرمادي تمثل مواقع العقد الحقيقية ضمن الشبكة، والعقدة باللون الأحمر هي العقدة الهجينة أما العدة التي على شكل دائرة تشير إليها 4 أسهم من الجهات الأربع هي عقدة المصرف.

خطوط الوصل السوداء القائمة تمثل مجالات اتصال العقد أي مثلاً بالنسبة للعقدة n2 فإن العقد الواقعة في مجالها هي n1 و 3n و 4n و s، أما العقدة n6 فلا يقع ضمن مجالها سوى n5 و HW100.

خطوط الوصل باللون الأحمر تربط العقد الحقيقية في المحاكى مع العقدة الهجينة وذلك من أجل الوصل مع البيئة الحقيقية خطوط الوصل هذه في الواقع تمثل الاتصال عن طريق كابل USB وليس لها علاقة بمجالات العقد.

2-5 آلية عمل الشبكة

الشكل التالي يوضح مخطط صندوقي يمثل الطريق من عقدة حقيقية الى عقدة المصرف:



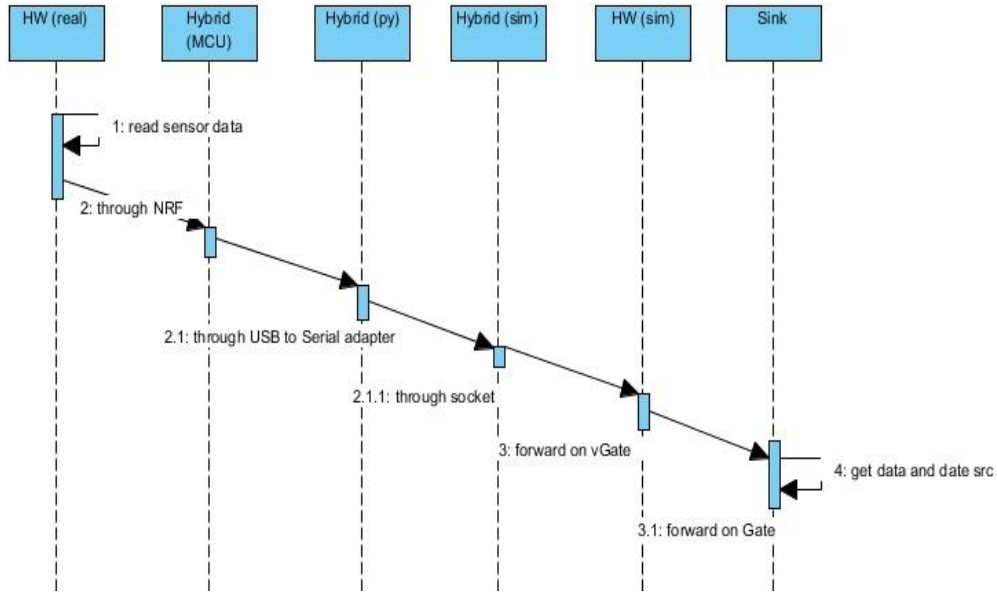
رسم توضيحي 9: مخطط صندوقي للطريق من عقدة حقيقية الى عقدة المصرف

في بداية المحاكاة تكون العقد الافتراضية والحقيقية في حالة استماع (تعمل كمستقبل فقط ولا ترسل) ويكون بارامتر عدد القفزات يساوي 255 = لانهاية، عندما يتم تشغيل المحاكاة تبتث عقدة المصرف رسالة تهيئة يكون فيها بارامتر عدد القفزات مساويا إلى 0 لكي تقوم بتبثيه العقد الواقعة ضمن مجالها الى بدء العمل. تقوم العقد التي تستقبل هذه الرسالة بقراءة تتحقق فيما إذا كانت هذه الرسالة مرسله اليها (عن طريق قراءة حقل الهدف من الرسالة المستقبلية) باعتبار ان الرسالة مرسله من المصرف فان عنوان الهدف هو عنوان البث العام ويساوي 255. وحيث لا توجد أي عقدة تملك هذا العنوان فان كل عقدة تصلها هذه الرسالة ستستدعي التابع handleOverHeard وتقوم بمقارنة بارامتر عدد القفزات لديها مع بارامتر عدد القفزات من الرسالة المستقبلية بعد زيادته بمقدار واحد ليصبح 1 في العقد الواقعة بالمستوى الأول (أي على بعد قفزة واحدة من المصرف). ففي حال كان أصغر يتم تعيين الأب لهذه العقدة (أي عنوان أصغر من 255) وتبدأ عندها بالعمل واخذ القراءات وارسال الرسائل وعندما تصل هذه الرسائل الى العقد في المستوى التالي تتكرر العملية السابقة ويكون بارامتر عدد القفزات الجديد هو 2 وهكذا.

بعد ان تبدأ الشبكة بالعمل وعندما تستقبل احدى العقد رسالة مرسله اليها (أي ان حقل الهدف من الرسالة المستقبلية مساو لعنوان تلك العقدة) يتم استدعاء التابع handleMyMessage والذي يقوم بقراءة بارامتر عدد الرسائل من الرسالة المستقبلية فإن كان أصغر من 5 (وهو العدد الأعظمي للرسائل التي يمكن تحميلها ضمن رزمة واحدة) تضيف العقدة رسالتها الخاصة المكونة من ثنائية (قراءة، عنوان المصدر) وترسل الرزمة الجديدة الى العقدة الأب. أما في حال كان عدد الرسائل يساوي 5 فإن العقدة المستقبلية تمرر الرزمة المستلمة كما هي ومن ثم تنشأ رزمة جديدة بعدد رسائل يساوي 1 تحوي قراءتها وعنوان المصدر ومن ثم ترسلها الى الأب.

بالنسبة للعقدة الهجينة Hybrid Node فإن آلية التواصل مع البيئة خارج المحاكي تتم عن طريق انشاء مخدم (server) يعمل على العنوان المحلي localhost برقم منفذ محدد يتم الاتصال به من خارج المحاكي عن طريق برنامج عميل تمت برمجته بلغة Python بحيث يتم التواصل بين المخدم والعميل وتبادل البيانات. برنامج العميل أيضا يقوم بالقراءة والارسال الى المتحكم عبر منفذ الـ UART وبالتالي يؤمن وسيلة التخاطب بين المحاكي والمتحكم PIC16F877A.

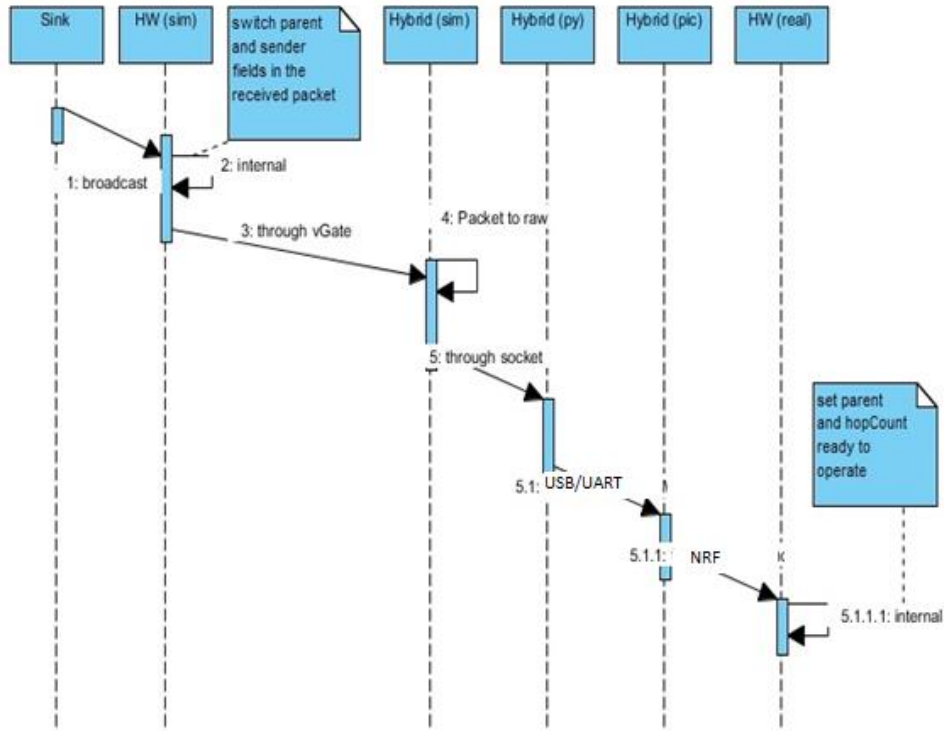
الشكل التالي يمثل عملية ارسال البيانات من العقدة الحقيقية وصولا الى المصرف:



رسم توضيحي 10: مخطط ارسال البيانات من العقدة الحقيقية الى المصرف

يظهر المخطط السابق أن العقدة الهجينة تتألف من قسمين (عتاد صلب وقسم برمجي) والقسم البرمجي يتألف من قسمين الأول يعمل في المحاكى على شكل مخدم والثاني هو عبارة عن برنامج العميل الذي تم كتابته بلغة Python. قسم العتاد الصلب هو عبارة عن دائرة ال Gateway. كما هو واضح في المخطط وعندما تريد العقدة الحقيقية HW Node ارسال البيانات فإنه في البداية تقوم بقراءة بيانات الحساس بعد ذلك تضمن هذه القراءة في رزمة وترسلها عبر ال NRF الى العقدة الهجينة التي تستلمها في ال NRF الخاص بها ومن ثم يتم قراءتها من قبل المتحكم وتميرها الى برنامج العميل الذي يعمل على الحاسب عن طريق وصلة UART/USB بعد ان يتلقاها العميل يمررها الى المحاكى عبر اتصال socket وعندما تتلقى العقدة الهجينة هذه الرزمة في المحاكى تمررها الى عقدة HW Node الموافقة عبر الوصلة المسماة vGate بعد ذلك تمرر هذه العقدة الرزمة الى عقدة المصرف أو الى العقدة التالية بحسب الرزمة عبر ال Gate .

أما الشكل التالي يوضح عملية بدء تهيئة الشبكة:



رسم توضيحي 8: مخطط ارسال البيانات من المصرف عند بداية عمل الشبكة

كما يوضح مخطط تهيئة الشبكة السابق فإن عمل الشبكة يتم تفعيله عن طريق عقدة المصرف التي تبدأ العمل بإرسال رسالة بث عام الى العقد الواقعة ضمن مجالها ويفرض ان أحد العقد الواقعة ضمن مجالها هي عقدة HW Node وبعد ان تتلقى هذه العقدة هذه الرزمة تمررها بعكس الطريق الموضح في الشكل الاول لتصل أخيرا إلى العقدة الحقيقية التي تقوم بضبط بارامترات الأب وعدد القفزات الخاصة بها.

يظهر في المخطط السابق عملية Packet to raw وهي عملية تتعلق بتحويل الأنواع بلغة ++C حيث ان الرزمة في المحاكي مؤلفة من حقول لها أنواع مختلفة عل سبيل المثال بارامتر عدد القفزات هو من النوع char وقراءة الحساس هي من النوع double وهكذا فإن عملية التحويل هذه تتم عبر التابع sendRawPacket الذي يقوم بتحويل هذه الأنواع المختلفة الى مصفوفة محارف string للإرسال عبر ال socket.

من أجل تسهيل عملية التعديل على البرنامج تم تقسيم الكود الى مكتبات مختلفة فمثلا تم وضع الكود الخاص ببروتوكول التوجيه في الملف المسمى routingProtocol الذي يحوي التوابع handleMyPacket و handleOverHeard وتعديل الكود الخاص بهذين التابعين يمكن تعديل طريقة عمل البروتوكول. في حال الرغبة بتغيير البروتوكول أساسا يمكن تعريف البروتوكول الجديد وتضمينه ضمن طبقة الشبكة ليصبح قابلا للاستخدام من قبل العقد في الشبكة.

الفصل السادس: تطورات مستقبلية

باعتبار أن نظم الاتصالات ونظم الشبكات في تطور مستمر، فإن الحاجة لإجراء تحسينات وتطويرات على أنظمة الاختبار والتصميم تكون كبيرة. وانطلاقاً من أهمية هذا المشروع في عملية تصميم الشبكات اللاسلكية فكان لابد من وضع بعض الافكار لمواكبة التطورات في هذا المجال.

من هذه الافكار امكانية اضافة بروتوكولات مختلفة أثناء عملية المحاكاة والمقارنة بين النتائج لاختيار البروتوكول الافضل، ومن الافكار الاخرى ايضا امكانية تحقيق تكنولوجيا انترنت الاشياء ومحاكاتها.

5-1 تعدد البروتوكولات في المحاكاة

ان امكانية تحقيق فكرة تعدد بروتوكولات التوجيه اثناء المحاكاة تقدم فائدة كبيرة وسرعة في عملية تصميم الشبكة ومن الافكار المقترحة لتحقيق ذلك امكانية وجود أكثر من نسخة لطبقة الشبكة يمكن تضمين النسخ ضمن الكود الخاص بالعقدة وبالتالي امكانية التبديل بين هذه النسخ وملاحظة النتائج.

5-2 انترنت الاشياء

وهو عبارة عن شبكة من الاجهزة التي تكون موجودة على السيارات أو الابنية أو قد تكون مرتبطة بالأشخاص أو مرتبطة بأي شيء يتطلب مراقبته أو التحكم به عبر شبكة الانترنت بحيث تقوم هذه الاشياء بجمع وتبادل المعلومات فيما بينها.

من الافكار المقترحة لمحاكاة انترنت الاشياء باستخدام هذا المشروع وصل عقدة المصرف (Sink) الموجودة ضمن المحاكاة مع الانترنت عبر مخدم يسمح بوصول المستخدمين من أي مكان في العالم الى المعلومات التي تقوم عقد الشبكة بإرسالها، وبالتالي يمكن انشاء نظام يقوم بوظائف معينة تبعا للمعلومات التي يحصل عليها عبر الانترنت من عقدة المصرف.

قمنا بتحقيق هذه الفكرة بشكل بسيط من خلال جعل عقدة المصرف عبارة عن مخدم يعمل على بروتوكول النقل UDP متصل مع برنامج مكتوب بلغة البايثون خارج المحاكاة ، يقوم برنامج البايثون بالقراءة من المصرف و يعمل من جهة أخرى كمخدم HTTP يمكن المستخدمين من الوصول الى معلومات اي عقدة و بالزمن الحقيقي عبر بروتوكول WebSocket .

قائمة بالملاحق

الملحق (أ): بروتوكول الوصل التسلسلي SPI

الملحق (ب): جدول تعليمات التحكم بجهاز الإرسال والاستقبال عبر الحاسب من خلال دائرة الربط

الملحق (ج): بروتوكول Enhanced Shockburst

الملحق (أ):

بروتوكول الوصل التسلسلي SPI

مقدمة عن الوصل التسلسلي:

إن طريقة الوصل التسلسلي تقوم بنقل المعلومات باستخدام سلك واحد حيث يتم ارسال البتات تباعا كل بت خلال زمن محدد وقد تم تطوير العديد من التقنيات والبروتوكولات في هذا المجال لتحقيق احتياجات النظم المضمنة، من هذه التقنيات USB، Ethernet، SPI، I2C، وتقنية الوصل التسلسلي التي تعتمد على بروتوكولات مثل RS232، RS485 ونميز نوعان من الوصل التسلسلي:

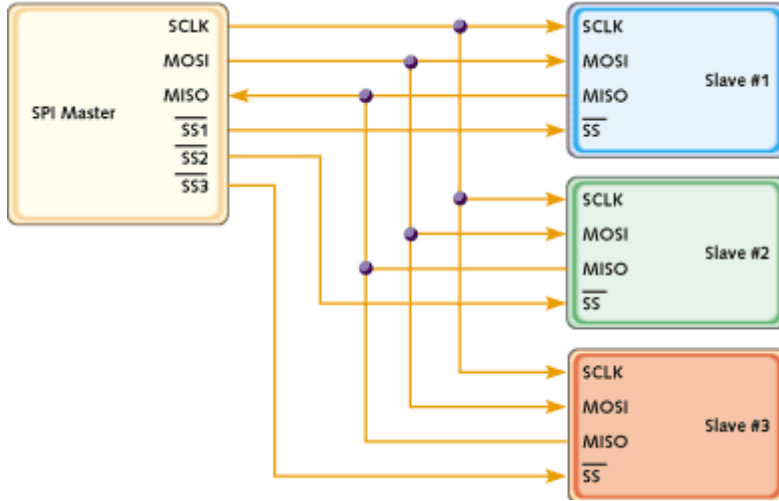
- الوصل الغير متزامن: وفيه يتم نقل المعلومات بشكل تسلسلي بدون استخدام اشارة نبضات ساعة خارجية ومن ميزات هذه الطريقة تخفيض عدد الاسلاك وعدد أطراف الدخول/الخروج في الدارات المستخدمة، لكن في المقابل علينا وضع بعض الشروط والقواعد التي تضبط عملية نقل المعلومات وتضمن وثوقيتها ومن هذه الشروط: تحديد بتات التزامن Synchronization bits، بتات فحص المساواة، وتحديد معدل البود.

- الوصل المتزامن: وفيه يتم ارسال اشارة المعلومات وفق اشارة نبضات ساعة خارجية مشتركة بين المرسل والمستقبل وفي هذه الحالة نتخلص من بعض عيوب الوصل الغير متزامن من اضافة بتات التزامن وتحديد معدل بود مشترك ومتفق عليه بين المرسل والمستقبل بالإضافة لتعقيد العتاد الصلب اللازم للقيام به، لكن في المقابل يلزمنا سلك ونهاية دخل/خروج مخصصة للإشارة الساعة.

أحد التقنيات المستخدمة في الوصل المتزامن هي تقنية SPI (Serial Peripheral Interface): واجهة الربط التسلسلي بين الاجهزة الطرفية والتي تعرف بال SPI، هي أحد التقنيات المستخدمة للربط بين الاجهزة في النظم المضمنة وذلك لمسافات قصيرة، تم تطوير هذه الواجهة من قبل شركة Motorola ثم أصبحت معيار معتمد. من اهم ميزات هذه التقنية هو امكانية نقل المعلومات في الاتجاهين في نفس الوقت (Full duplex) بالاعتماد على هيكلية السيد-العبد (Master-Slave)، حيث يمكن ان يكون هناك أكثر من عبد لكن فقط سيد واحد. ويكون لكل عبد خط مخصص له يدعى (Slave Select) SS منفصل.

- عند ارسال الاوامر من السيد تصل الاوامر الى كل عبد متصل معه ولكن لا يتم تنفيذها إلا من قبل العبد قيد التنفيل (عادة بجعل SS جهد منخفض).

- يظهر الشكل التالي أحد اشكال الوصل في SPI:

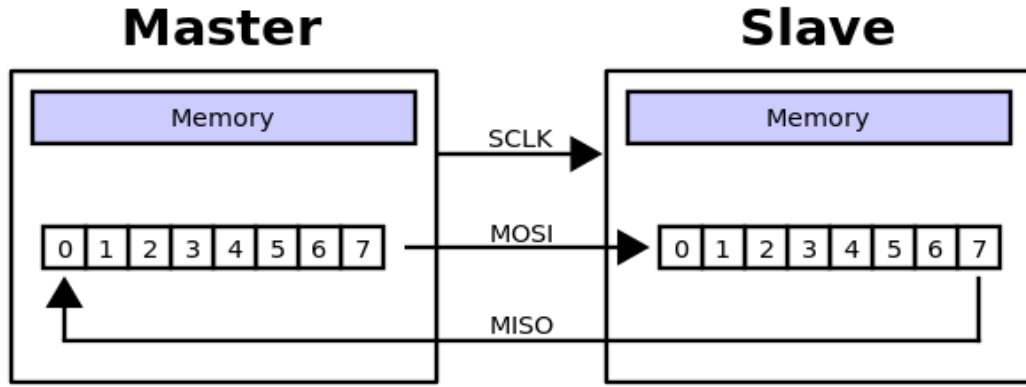


رسم توضيحي 11: وصل SPI سيد واحد وعدة عبيد بطريقة الوصل التقليدية باستخدام slave select

- الخطوط اللازمة لتحقيق SPI:
- SCLK: إشارة نبضات الساعة و التي تخرج من السيد الى العبد.
- MOSI (Master Output Slave Input) كما يظهر الاسم، طرفية نقل المعلومات من السيد الى العبد.
- MISO (Master Input Slave Output): عكس الطرفية السابقة، طرفية نقل المعلومات من العبد الى السيد.
- SS (Slave Select): طرفية اختيار العبد.
- قد تختلف تسميات هذه الطرفيات بشكل طفيف من دائرة الى اخرى.

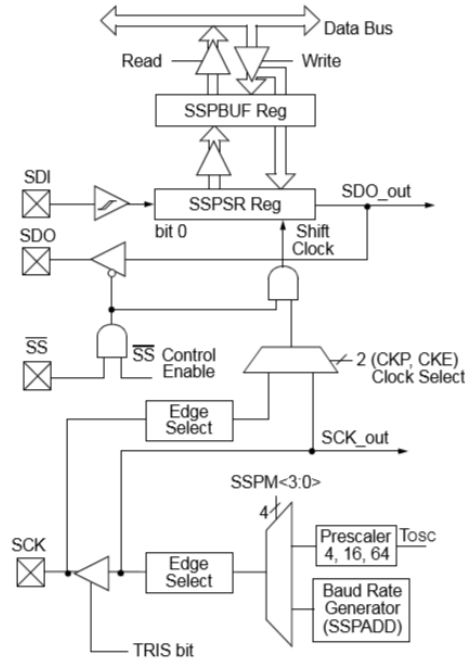
نقل المعطيات في SPI:

- للقيام بعملية نقل المعطيات يقوم السيد بشكل عام بالمهام التالية:
- تحديد تردد الساعة (عادة في مجال ال MHz)، كما يتم ضبط كل من السيد والعبد على العمل على نفس قطبية لساعة وعلى نفس الجبهة (هابطة أو صاعدة).
- تحديد العبد الذي يتم التعامل معه وذلك عن طريق الخط SS، وفي حال وصل السيد مع أكثر من عبد ولكل منهم تهيئة مختلفة يكون السيد في حاجة لان يعيد ضبط نفسه قبل العمل مع العبد الجديد.
- خلال عملية نقل المعطيات يحدث نقل بالاتجاهين بين السيد والعبد كما يوضح الشكل التالي:



رسم توضيحي 12: تبادل البيانات في SPI بين السيد والعبد

- عملية الارسال تتضمن مسجلي ازاحة واحد في السيد والآخر في العبد يكون عادة عرض كل منهما 8 بت.
- عند عملية الارسال يتم ازاحة بت واحد في كل مرة (مع كل نبضة ساعة) مع ازاحة البت الاكثر اهمية أولاً. وفي نفس الوقت تم ادخال بت جديد إلى خانة البت الاقل اهمية.
-
- بعد ان يتم ازاحة كل من البتات الثمانية سيكون كل من السيد والعبد قد تبادلا قيم مسجلاتها وفي حال وجود المزيد من البيانات التي تتطلب الارسال يتم تحميل البايت الجديدة في مسجل الازاحة ويتم تكرار العملية السابقة.
- هناك ثلاث سيناريوهات محتملة بالنسبة للعملية السابقة:
- السيد يرسل بيانات مفيدة والعبد يرسل بيانات لا قيمة لها.
- السيد يرسل بيانات مفيدة والعبد يرسل بيانات مفيدة.
- السيد يرسل بيانات لا قيمة لها والعبد يرسل بيانات مفيدة.
- عند انتهاء عملية الارسال يوقف السيد اشارة نبضات الساعة ويلغي تفعيل العبد المختار عن طريق خط التحكم SS.
- واجهة المحيطيات التسلسلية SPI في المتحكم PIC من شركة Microchip:
- تتيح الوحدة MSSP (Master Synchronous Serial Port) في المتحكم pic12LF1552 استخدام ال SPI، ويظهر الشكل التالي المخطط الصندوقي ل SPI :



رسم توضيحي 13: وحدة SPI ضمن المتحكم

المسجلات التي يجب ضبطها للدخول في وضع الـ SPI :

1- (SSPSTAT) MSSP1 status register يتم فيه ضبط البتات التالية :

- <7> SSPSTAT: SMP: يتم من خلاله تحديد مرحلة اعتيان البيانات الداخلة وذلك اما في منتصف البت (إذا كانت قيمته SMP=0) او في اخر البت (SMP=1).
- <6> SSPSTAT: CKE: من خلاله يتم تحديد لحظة ارسال البت, وذلك اما عند الانتقال من وضع الخمول الى الوضع النشط (CKE=0) أو بالعكس.

1 (SSPCON1) MSSP1 control register يتم فيه ضبط البتات التالية :

- <5> SSPCON1: SSPEN: من خلال هذا البت يتم تفعيل عمل الوحدة MSSP وذلك عند القيمة 1.
- <4> SSPCON1: CKP: تحديد قطبية الساعة في حالة الخمول حيث تكون حالة الخمول منخفضة عند القيمة 0.
- <3:0> SSPCON1: SSPM: يتم من خلال هذه البتات تحديد اذا كان المتحكم في وضع السيد أو العبد, وبالتالي اذا كان السيد يتم تحديد تردد الساعة, اما اذا كان عبد يتم من خلالها تفعيل او ايقاف طرفية اختيار العبد SS.

2- (SSPCON3) MSSP1 control register 3 يحوي البتات التالية :

- <4> SSPCON3: BOEN: يستخدم هذا البت في وضع العبد فقط وفيه يتم تحديد اذا ما كان سيتم الكتابة في المسجل SSPBUF بغض النظر عن اذا كان قد تم قراءته أم لا (BOEN=1), اما اذا كان BOEN=0 و كان المسجل SSPBUF لم يقرأ بعد (البت BF من المسجل SSPSTAT يكون مرتفعاً اذا لم تتم قراءة SSPBUF) فان المعلومات الجديدة لا يتم تحميلها الى المسجل SSPBUF و يرتفع البت SSPOV في المسجل SSPCON1.

- MSSP1 data buffer register (SSPBUF) : من خلال هذا المسجل يتم قراءة و كتابة مسجل الازاحة
- MSSP1 shift register (SSPSR).

إذا يلزمنا لتهيئة ال SPI ضبط البتات SSPCON1<0:5> و SSPSTAT<6:7>، وذلك لتحديد ما يلي:

- نمط السيد (SCK هي خرج).
 - نمط العبد (SCK دخل).
 - قطبية اشارة الساعة SCK في وضع الخمول.
 - مرحلة اخذ العينة للبيانات المستقبلية (في منتصف او نهاية البت المستقبل).
 - اخراج البيانات عند الحافة الصاعدة أو الهابطة.
 - معدل اشارة نبضات الساعة وذلك في وضع السيد.
 - تشغيل او ايقاف خاصية اختيار العبد وذلك في وضع العبد فقط.
- إن وحدة MSSP1 تتألف من مسجل ازاحة يستخدم للإرسال والاستقبال SSPSR ومسجل عبارة عن عازل Buffer SSPBUF. يقوم SSPSR بإزاحة البيانات باتجاه الدخل أو الخرج بداية من البت الأكثر اهمية، في حين يقوم SSPBUF بتخزين البيانات التي تم كتابتها إلى SSPSR. عند الانتهاء من استقبال البتات الثمانية يتم نقل البايت المتشكل منها إلى SSPBUF ومن ثم يتم وضع قيمة واحد منطقي في البت BF (BUFFER FULL) من المسجل SSPSTAT. ويرتفع علم المقاطعة SSPIF. وجود كلا المسجلين SSPBUF و SSPSR يسمح بالبدء باستقبال البت التالي قبل قراءة البت الحالي، سيتم تجاهل اي عملية كتابة الى SSPBUF أثناء عملية الارسال أو الاستقبال وسيترفع البت WCOL من المسجل SSPCON1 والذي يعني حصول تضارب اثناء الكتابة. يتم تصفير هذا البت من خلال برنامج المستخدم من اجل السماح بالقيام بعملية الكتابة التالية الى المسجل SSPBUF. عند استقبال البيانات يجب ان يتم قراءة المسجل SSPBUF قبل انتهاء كتابة البايت في المسجل SSPSR. عند قراءة SSPBUF سيتم تصفير البت BF.

يمكن استخدام المقاطعات الخاصة بالوحدة MSSP1 من اجل تحديد انتهاء استقبال او ارسال البيانات.

في حال عدم استخدام المقاطعة يجب على برنامج المستخدم ان يقوم بالتحقق من البتات الموافقة لتجنب حدوث تضارب.

في وضع السيد:

مكن للسيد ان يبدأ نقل البيانات في اي وقت، ويحدد متى يمكن للعبد ان يقوم بعملية النقل، وذلك لأنه يتحكم بنبضات الساعة عبر الخط SCK.

في وضع السيد يتم ارسال او استقبال البيانات بنفس اللحظة التي يتم فيها الكتابة الى المسجل SSPBUF.

في حال كانت وظيفة الجهاز هي الاستقبال فقط يمكن ايقاف الخط SDO عن طريق برمجته كدخل. وستستمر عملية الاستقبال كما في السابق.

في وضع السيد يمكن برمجة تردد نبضات الساعة ليكون أحد الخيارات التالية:

- FOSC/4 (or TCY)
- FOSC/16 (or 4 * TCY)
- FOSC/64 (or 16 * TCY)

$$- \text{FOSC}/(4 * (\text{SSPADD} + 1))$$

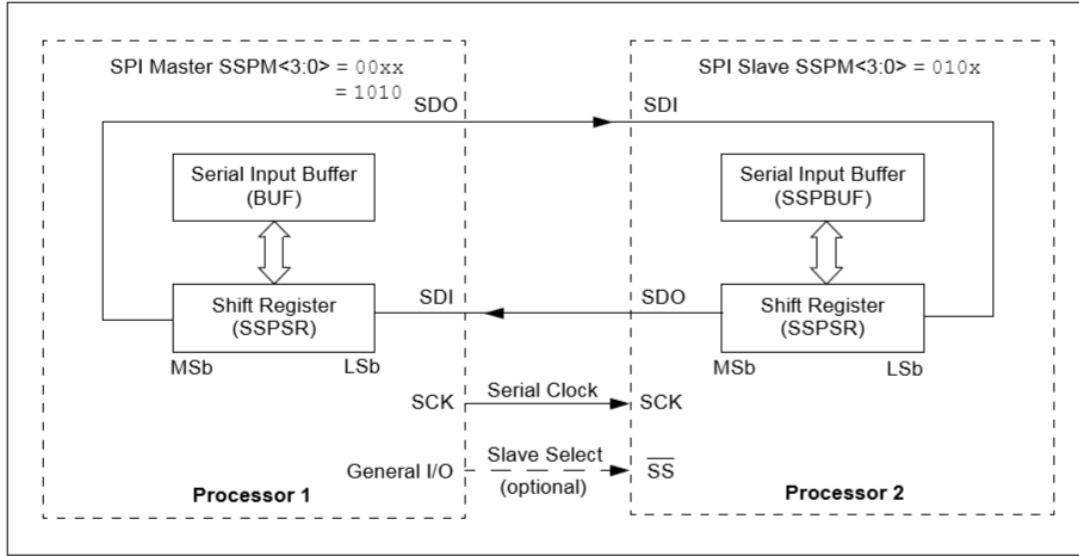
وضع العبد:

في وضع العبد يتم ارسال واستقبال البيانات عند وصول نبضات الساعة القادمة من السيد عبر الخط SCK

عند نقل اخر بت فان بت المقاطعة SSPIF سيرتفع.

قبل تفعيل وحدة الـ SPI في وضع العبد يجب ضبط نظام العمل بنبضات الساعة ليكون متوافقا مع السيد.

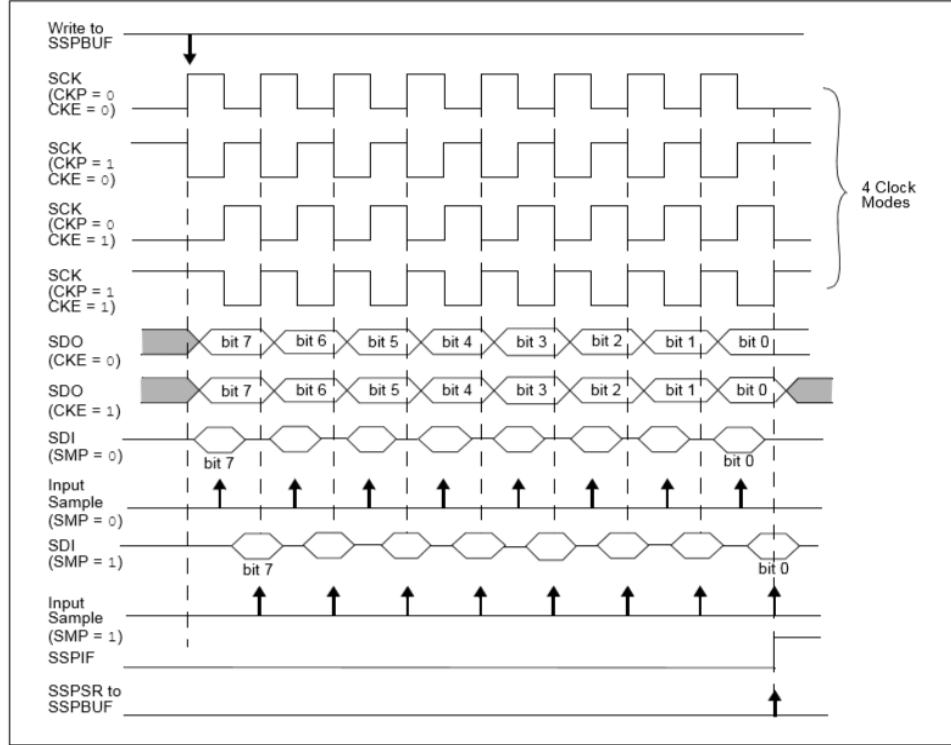
يوضح الشكل التالي مثال عن ربط معالجين معا عبر الـ SPI :



رسم توضيحي 14: ربط متحكمين عبر SPI

كما يوضح المخطط التالي المخططات الزمنية لعمل الـ SPI وفق اوضاع مختلفة:

FIGURE 18-6: SPI MODE WAVEFORM (MASTER MODE)



رسم توضيحي 15: المخططات الزمنية SPI-1 وفق أوضاع مختلفة

مميزات ال SPI:

- يؤمن سرعات أعلى من نظام الاتصال التسلسلي غير المتزامن.
- بسبب انه يتم قذح البتات مع نبضات الساعة مع امكانية تخصيص القطبية والجهة يمكن ان يكون الطرف المستقبل عبارة عن مسجل ازالة.
- يمكن وصل أكثر من عبد.

مساوئ SPI:

- يتطلب خطوط اشارة أكثر من طرق التوصيل الاخرى I2C مثلا.
- الحاجة إلى ضبط كلا الطرفين (المرسل والمستقبل) مسبقا.
- السيد يتولى جميع عمليات الاتصال (اجهزة العبد غير قادرة على التواصل فيما بينها بشكل مباشر).
- الحاجة إلى خط SS لكل عبد ما يتطلب عدد كبير من الخطوط في حال وجود عدد كبير من الاجهزة العبد.
- إن نظام SPI لا يستخدم آلية الاخطار بوصول البيانات Acknowledgement ولا يتيح آليات للتحكم بتدفق المعطيات.

الملحق (ب):

جدول تعليمات التحكم بجهاز الارسال والاستقبال عبر الحاسب من خلال دارة الربط

جدول 8: تعليمات التحكم بالـ NRF24L01 عن طريق الحاسب

المحرف الموافق	have_result	البرامترات args	الأمر command	العملية
1	0	arg1= Data to be written	SPI_WRITE	كتابة على الـ NRF عبر SPI
2	1	arg1= Dummy Data ₁	SPI_READ	قراءة من الـ NRF عبر SPI
3	0	None ²	CE_PIN_LOW	خفض الـ CE_PIN
4	0	None	CE_PIN_HIGH	رفع الـ CE_PIN
5	0	None	TOGGLE	قدح CSN_PIN مع خفض CE_PIN
6	0	None	TOGGLE_CSN	قدح CSN_PIN دون خفض CE_PIN
a	0	None	INIT_SPI	تهيئة وحدة SPI في المتحكم
b	1	None	GET_STATUS	قراءة المسجل STATUS ضمن الـ NRF
c	0	arg1=W_REGISTER + 'REGISTER' ³ arg2= Data	NRF_RW	الكتابة إلى مسجلات الـ NRF
c	1	arg1= 'REGISTER' ³ arg2= Data ⁴	NRF_RW	القراءة من مسجلات الـ NRF
d	0	arg1 = 0 => power down else => power up	POWER_MODE	وضع الطاقة
e	0	arg1= 2 for 2Mbps else for 1Mbps	SET_DATA_RATE	ضبط معدل الارسال
f	0	arg1= channel number from 0 to 125	SET_RF_CH	اختيار القناة

g	0	arg1= 1 for 1 Byte CRC 2 for 2 Byte CRC	CRC_CONFIG	اختيار عدد بايتات الـ CRC
h	0	arg1= number of pipe from 0 to 5	EN_DYNAMIC_PAYLOAD	تفعيل طول بيانات متغير على الـ pipe المطلوب
i	0	arg1= address width from 3 to 5	SET_ADDRESS_WIDTH	ضبط طول العنوان
j	0	arg1= pipe number arg2= address width arg3= address	SET_PIPE_ADDRESS	ضبط عنوان الـ pipe
k	0	None	RX_MODE	وضع الـ NRF في وضع الاستقبال
l	0	arg1= 0 for low gain 1 for high gain	RX_LNA_GAIN	تحديد ربح مضخم الدخل في المستقبل
m	0	arg1= pipe number arg2= data width arg3= data	RX_ACK_WITH_PAYLOAD	إضافة بيانات إلى حزمة الـ Acknowledgment
n	0	arg1= pipe register ⁵ arg2= length	RX_SET_STATIC_PAYLOAD	ضبط طول حمل ثابت
o	0	None	TX_MODE	وضع الـ NRF في وضع الإرسال
p	0	arg1=data width arg2=data	W_TX_BUFFER	الكتابة إلى TX BUFFER
q	0	arg1=delay arg2=retransmit count ⁶	TX_AUTO_RETRANSMIT	ضبط عملية إعادة الإرسال
r	0	arg1= pipe number	TX_DISABLE_AUTO_ACK	تعطيل Auto Ack
s	0	arg1=pipe number	TX_ENABLE_AUTO_ACK	تفعيل Auto Ack
t	0	arg1= address width arg2= address	TX_SET_ADDRESS	ضبط عنوان الإرسال
u	0	arg1= 0 to 4 0 = lowest power 4 = highest power	TX_OUTPUT_POWER_CONTROL	ضبط طاقة الإرسال
v	0	None	RX_DEFAULT	وضع RX باعدادات افتراضية
w	0	None	TX_DEFAULT	وضع TX باعدادات الافتراضية

الملحق (ج):

بروتوكول Enhanced Shockburst

مقدمة عن Enhanced ShockBurst(ESB):

ESB هو بروتوكول طبقة نقل المعطيات Data Link layer في سلسلة الاجهزة nRF24LXX من شركة Nordic Semiconductor, يدعم هذا البروتوكول نقل المعطيات بالاتجاهين كما يدعم التأكد على وصول الحزمة (packet acknowledgement) و يدعم ايضا اعادة بث الحزم الضائعة, و من ميزاته ايضا سهولة استخدامه من قبل المتحكمات الصغيرة. ESB يمتلك خاصية تبادل المعطيات بالاتجاهين بشكل موثوق في طبقة نقل المعطيات Data link layer, يتم تبادل الحزم بين نقطتين كل منهما عبارة عن جهاز مرسل/مستقبل (Transceiver), يلعب احد الجهازين دور المستقبل الرئيسي (PRX (Primary Receiver), والجهاز الاخر دور المرسل الرئيسي (PTX (Primary Transmitter).

المواصفات الرئيسية لبروتوكول ESB:

- طول ديناميكي للمعطيات من 1 حتى 32 بايت.
- التعامل مع الحزم بشكل اوتوماتيكي.
- ارسال اشعار استقبال المعطيات، وإعادة الارسال بشكل اوتوماتيكي.
- 6 مسارات للمعطيات (data pipes) بين الاجهزة بشكل شبكة نجمية.

يعتمد البروتوكول Enhanced Shockburst على تكنولوجيا تدعى Shockburst من شركة Nordic وهذه التكنولوجيا موجودة في سلسلة nRF240X وكان هدف ال Shockburst هو وضع ما أمكن من العمليات على المعطيات المرسل والمستقبل داخل شريحة ال NRF مع اعطاء مجال من الحرية والمرونة للمستخدم. يستخدم ال ESB تقنية Shockburst في التعامل مع الحزم بشكل اوتوماتيكي، حيث يقوم ال Shockburst في عملية الارسال بتجميع الحزمة ووضعها في الجزء المخصص للإرسال، اما في عملية الاستقبال يقوم ال ShockBurst بالبحث وبشكل مستمر عن عنوان صحيح في الحزم التي يستقبلها، وعند ايجاد عنوان صحيح للحزمة يقوم باستقبال كامل الحزمة والتحقق من صحتها باستخدام كود التحقق CRC، فاذا كانت الحزمة صالحة وخالية من الاخطاء يتم نقلها الى المسجل RX FIFO. اما ال ESB فيتيح عملية ارسال المعطيات بشكل موثوق من خلال خاصية ارسال اشعار باستقبال البيانات، كما يمكن تحقيق اتصال بالاتجاهين من خلال اضافة معلومات الى هذا الاشعار.

ارسال واستقبال الاشعار في ESB:

تحدث عملية ارسال واستقبال الاشعار وفق شكل التالي:

تبدأ عملية الاتصال بقيام المستخدم بإرسال المعلومات من المرسل الرئيسي PTX الى المستقبل الرئيسي PRX، يقوم بعدها ال ESB وبشكل اوتوماتيكي بوضع ال PTX بوضع الاستقبال لكي يستقبل اشعار وصول المعلومات ack.

إذا تم استقبال المعلومات بشكل صحيح من ال PRX يقوم ال ESB بشكل اوتوماتيكي بإرسال اشعار الاستلام ack الى ال PTX ثم يعود الى وضع الاستقبال.

إذا لم يستقبل ال PTX الاشعار خلال زمن محدد، يقوم بإعادة الارسال بشكل اوتوماتيكي واعادة الخطوات السابقة.

ولتحقيق الاتصال ثنائي الاتجاه يمكن لل PRX ارفاق معلومات ضمن اشعار الاستقبال ack.

شكل حزمة ال ESB (enhanced shockburst packet format):

يبين الشكل التالي رزمة ال ESB:

Preamble 1 byte	Address 3-5 byte	Packet Control Field 9 bit	Payload 0 - 32 byte	CRC 1-2 byte
-----------------	------------------	----------------------------	---------------------	--------------

رسم توضيحي 16: شكل رزمة ال ESB

حقل المقدمة (preamble):

وهي مجموعة من البتات تستخدم لتحديد المستويات المنطقية في المستقبل، وهي في ال ESB بطول 1 بايت وتكون اما 01010101 وذلك إذا كان البت الاول في حقل العنوان هو 0، او تكون 10101010 إذا كان البت الاول في حقل العنوان 1، والهدف من ذلك هو استقرار المستقبل بإعطائه العدد الاكبر من الانتقالات بين المستويات المنطقية.

حقل العنوان (address):

هذا الحقل مخصص لعنوان المستقبل، يمكن تعديل حجم هذا الحقل من خلال المسجل AW، حيث يمكن ان يكون العنوان 3 او 4 او 5 بايت.

ملاحظة: ان اختيار عناوين حيث يوجد انتقال واحد فقط لمطال الاشارة (مثل FFFFFFFF000) قد يسبب أخطاء في الاستقبال حيث يمكن ان يتم كشف هذا العنوان في اشارات الضجيج الغير مرغوبة.

حقل التحكم بالحزمة (packet control field):

يقسم هذا الحقل الى ثلاثة حقول فرعية كما يظهر الشكل:

Payload length 6bit	PID 2bit	NO_ACK 1bit
---------------------	----------	-------------

رسم توضيحي 17: حقل التحكم بالحزمة لل ESB

الحقل الاول مخصص لتحديد طول البيانات المرسله، يتراوح طول البيانات المرسله من 0 حتى 32 بايت، في حالة ال 0 بايت (عدم ارسال بيانات) يكون الحقل Payload length=000000 وتستخدم هذه الحالة عند ارسال اشعار الوصول ack، اما القيمة Payload=100000 فهي تمثل 32 بايت، واي زيادة على القيمة السابقة تهمل.

ملاحظة: يستخدم الحقل Payload length عند تفعيل الطول الديناميكي للبيانات.

الحقل الثاني PID (packet identity) ويستخدم لمعرفة إذا كانت الحزمة المستقبلية جديدة أو معاد إرسالها، وبالتالي يمنع الPRX من تقديم نفس الحزمة إلى المتحكم الصغري أكثر من مرة.

يقوم المرسل بزيادة قيمة هذا الحقل مقدار واحد مع كل حزمة يتم استقبالها عبر ال SPI. في حالة ضياع أكثر من حزمة في وسط الإرسال فإن قيمة حقل PID للحزمة الجديدة قد تصبح مساوية لقيمة PID للحزمة القديمة . في هذه الحالة سيقوم المستقبل بمقارنة قيمة الحقل CRC لكل من الحزمتين الجديدة والقديمة (اللتين لهما نفس قيمة ال PID) في حال تساوي قيمتي حقل ال CRC في كل من الحزمتين عندها فإن الحزمة الجديدة تعتبر مكررة ويتم تجاهلها.

الحقل NO_ACK من خلاله يتم تحديد إذا كان المرسل يريد استقبال اشعار ام لا، حيث ان اعطاؤه القيمة 1 تخبر المستقبل بعدم ارسال الاشعار.

ملاحظة: الحقل NO_ACK يستخدم فقط عند تفعيل الميزة ارسال الاشعار التلقائي Auto-acknowledgment .feature

البيانات المرسله (payload): يحتوي هذا الحقل على البيانات المراد إرسالها، ويمكن ان تكون بطول 0 حتى 32 بايت ويتم إرسالها دون اي تعديل.

كود التحقق (CRC): وهو الية التحقق من الاخطاء في الحزمة، ويمكن ان يكون 1 او 2 بايت، ويحسب اخذا بعين الاعتبار الحقول Address، Packet Control Field، Payload .

في حالة استخدام CRC 1-Byte فإن كثير الحدود المستخدم هو: $X^8 + X^2 + X + 1$

في حالة استخدام CRC 2-Byte فإن كثير الحدود المستخدم هو: $X^{16} + X^{12} + X^5 + 1$

يتم تحديد عدد البايتات في حقل CRC عن طريق البت CRCO في المسجل CONFIG.

يتم رفض كل حزمة مستقبلية في حال فشل اختبار CRC.

التعامل الاوتوماتيكي مع الحزم:

1- تعيين حجم البيانات المرسل:

يتيح ال ESB اليتين لتحديد حجم البيانات، ديناميكية وستاتيكية. في الحالة الافتراضية يكون حجم البيانات ثابت (ستاتيكي)، ويتم تحديد حجم البيانات من المسجل RX_PW_Px في جانب الاستقبال، اما في الارسال فان حجم البيانات يعرف من خلال عدد البايتات التي تدخل الى مسجل الارسال TX_FIFO، وبالتالي يجب ان يكون حجم البيانات في TX_FIFO مساو للحجم المحدد في المستقبل في المسجل RX_PW_Px.

الالية الثانية لتحديد حجم البيانات هي الالية الديناميكية، وفيها يمكن للمستقبل وبحرية مطلقة ارسال بيانات بأحجام مختلفة الى المستقبل، وفي هذه الحالة يقوم المستقبل وبشكل اوتوماتيكي بقراءة الحقل Payload length ومعرفة حجم البيانات، ولا يحتاج لقراءة المسجل RX_PW_Px، ويمكن للمتحكم الصغري ان يقرأ حجم البيانات عن طريق التعليمات R_RX_PL_WID.

لتفعيل الالية الديناميكية يجب تفعيل البت EN_DPL في المسجل FEATURE، كما يجب تفعيل البت المناسب في المسجل DYNPD في المستقبل، ويجب تفعيل البت DPL_P0 في المسجل DYNPD في وضع الارسال. ملاحظة: في حالة كان طول الحمولة أكبر من 32 بايت فهذا يعني وجود خطأ في الحزمة ويجب أن تهمل عن طريق الأمر Flush_RX.

2- تجميع الحزمة بشكل اوتوماتيكي (Automatic packet assembly):

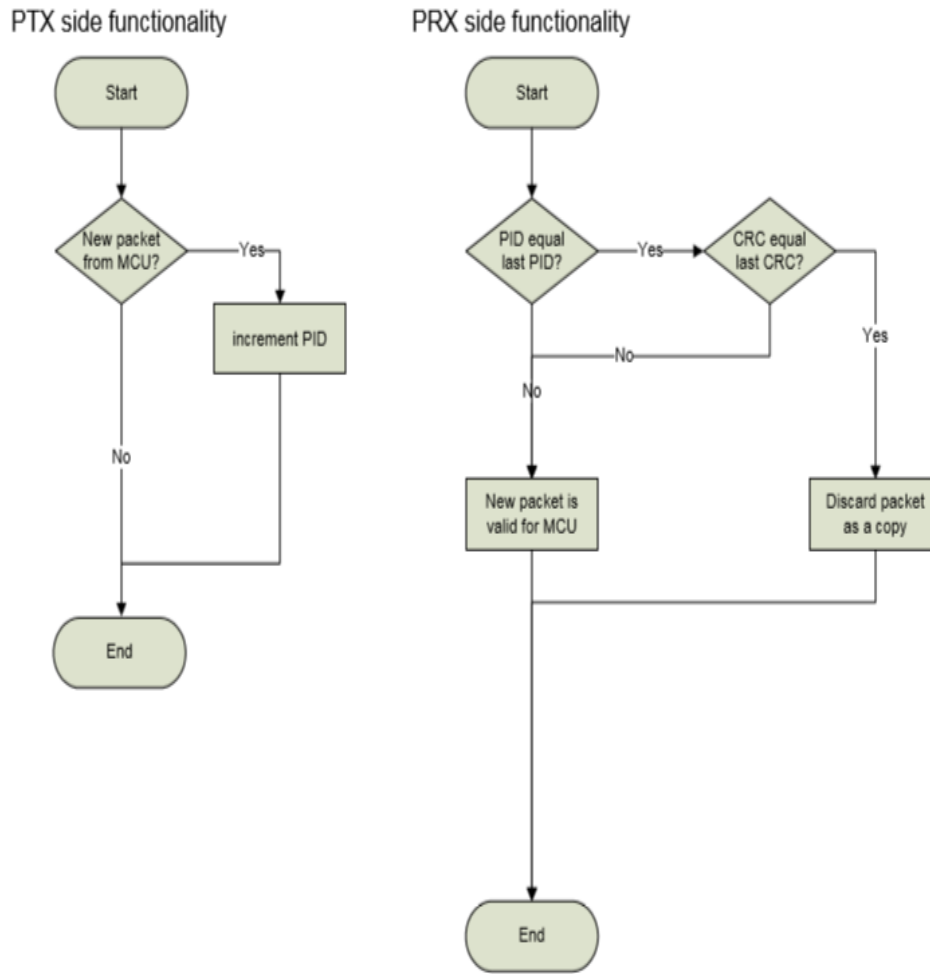
من ميزات ال ESB هو تجميع الحزمة بشكل اوتوماتيكي، حيث يتم تجميع الحقول preamble , address , packet control field , payload , CRC.

ان الحقل preamble يتم توليده بشكل اوتوماتيكي بحسب الحقل address كما ذكر سابقا.

الحقل Address يتم ضبطه من خلال المسجل TX_ADDR، ويتم ضبط طول العنوان عبر المسجل AW.

الحقل Packet control: يتم ضبط الحقل الفرعي Payload length بشكل الي عند اختيار الالية الديناميكية لحجم البيانات، وذلك بحسب البايتات الموجودة في المسجل TX_FIFO، اما في حال الالية الستاتيكية فلا يتم استخدام هذا الحقل.

الحقل الفرعي PID يتم زيادته في كل مرة يتم فيها توليد حزمة جديدة، وفي حالة اعادة الارسال لا يتم زيادته، ويظهر المخطط التالي ذلك:

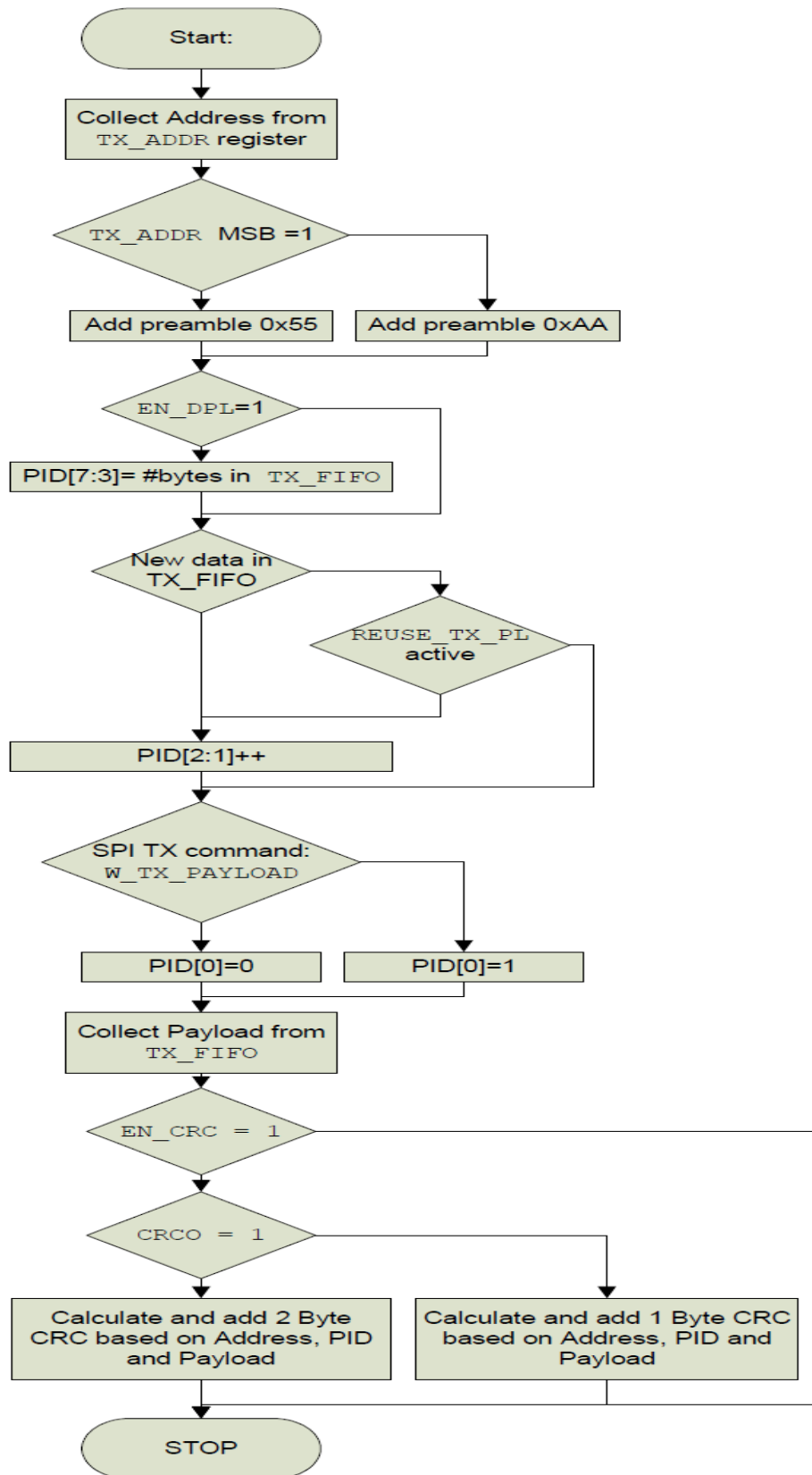


رسم توضيحي 18: آلية زيادة الحقل PID في رزمة ال ESB

الحقل الفرعي NO_ACK يمكن التحكم به من خلال الامر W_TX_PAYLOAD_NOACK، وذلك بعد تفعيل البت EN_DYN_ACK في المسجل FEATURE. وعند ابطال ال ack لا يقوم المستقبل بإرسال اشعار الاستلام، ولا يدخل المرسل وضع الاستقبال منتظرا الاشعار.

الحقل CRC يتم توليده بشكل الي ولكن يمكن تحديد حجم هذا الكود من خلال ضبط البت CRCO في المسجل .CONFIG

المخطط التدفقي التالي يوضح عملية التركيب التلقائي للرزمة:

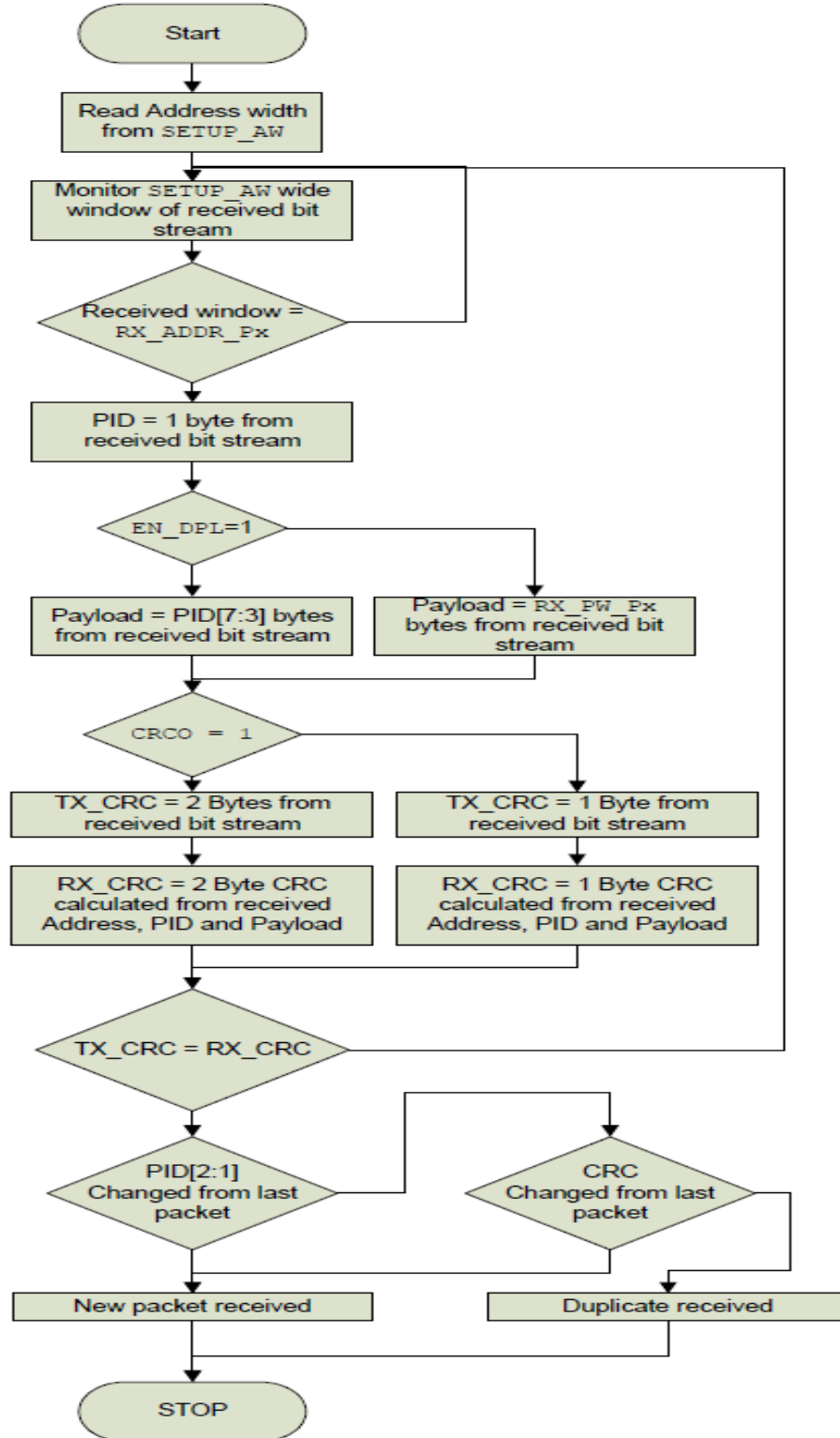


رسم توضيحي 19: التركيب التفصيلي للرمز في الـ NRF24L01

-تفكيك الحزمة بشكل أوتوماتيكي (Automatic packet assembly):

بعد أن يتم التحقق من سلامة الحزمة المستلمة يقوم ESB بتفكيك الحزمة المستقبلية ويضع الحمولة (البيانات المرسله) في RX_FIFO ويرتفع العلم RX_DR.

الشكل التالي يوضح كيفية تفكيك الحزمة:



خاصية تبادل الحزم وتسليمها بشكل تلقائي:

يتيح ESB امكانيتين للتعامل التلقائي مع الحزم وهما:

الاشعار التلقائي باستلام الحزمة: Auto Acknowledgment

يتيح الاخطار التلقائي باستلام الحزمة الامكانية للمستقبل بإعلام المرسل باستلام حزمة صالحة تلقائيا ودون تدخل من المتحكم.

يتم تفعيل هذه الميزة عن طريق المسجل EN_AA.

ملاحظة: إذا كان EN_AA مفعّل، وحقل NO_ACK مفعّل أيضا لا يتم ارسال اشعار.

يمكن ان تحتوي حزمة الاخطار على معلومات يخصصها المستخدم (حيث ان استخدام هذه الميزة يتطلب تفعيل طول بيانات ديناميكي)، ويتم ذلك من خلال قيام المستقبل بوضع المعلومات المراد ارفاقها مع الاشعار ضمن المسجل TX_FIFO وذلك من خلال التعليمة W_ACK_PAYLOAD، حيث يتم ارسال هذه المعلومات عند استقبال حزمة من المرسل، ويمكن للمستقبل وضع ثلاث احمال (payloads) في الانتظار بنفس الوقت.

إعادة الارسال التلقائية: Auto Retransmission

تتم اعادة الارسال التلقائية في حال تم ارسال حزمة ولم يتم استلام اخطار بوصول هذه الحزمة إلى وجهتها.

يمكن تخصيص عدد مرات محاولة اعادة الارسال والازمنة بين هذه المحاولات عن طريق المسجل .SETUP_RETR.

يمكن ايضا ضبط الزمن بين انتقال المرسل الى وضع الاستقبال منتظرا الاشعار واعادة الارسال من خلال المسجل .SETUP_RETR.

ملاحظة: عند استخدام حزمة اخطار مع حمولة مخصصة من قبل المستخدم يجب الانتباه الى الزمن بين محاولات اعادة الارسال بحيث لا يكون أصغر من (زمن الانتقال لوضع الارسال + وقت انتقال الحزمة في وسط الارسال).

مثلا من أجل معدل ارسال 1Mbps وعنوان بطول 5 Byte لا يجب استخدام حمولة أكبر من 5 Byte ضمن حزمة الاخطار عند زمن تأخير لإعادة الارسال 250us.

تستمر محاولات اعادة الارسال حتى أن يتم استلام الاخطار أو ان ينتهي عدد المحاولات المخصص.

هناك عدادان ضمن المسجل OBSERVE_TX يزدادان مع في كل مرة يحدث ضياع للحزمة المرسل. العداد الاول هو ARC_CNT والذي يقوم بعد محاولات اعادة الارسال من اجل الحزمة الحالية.

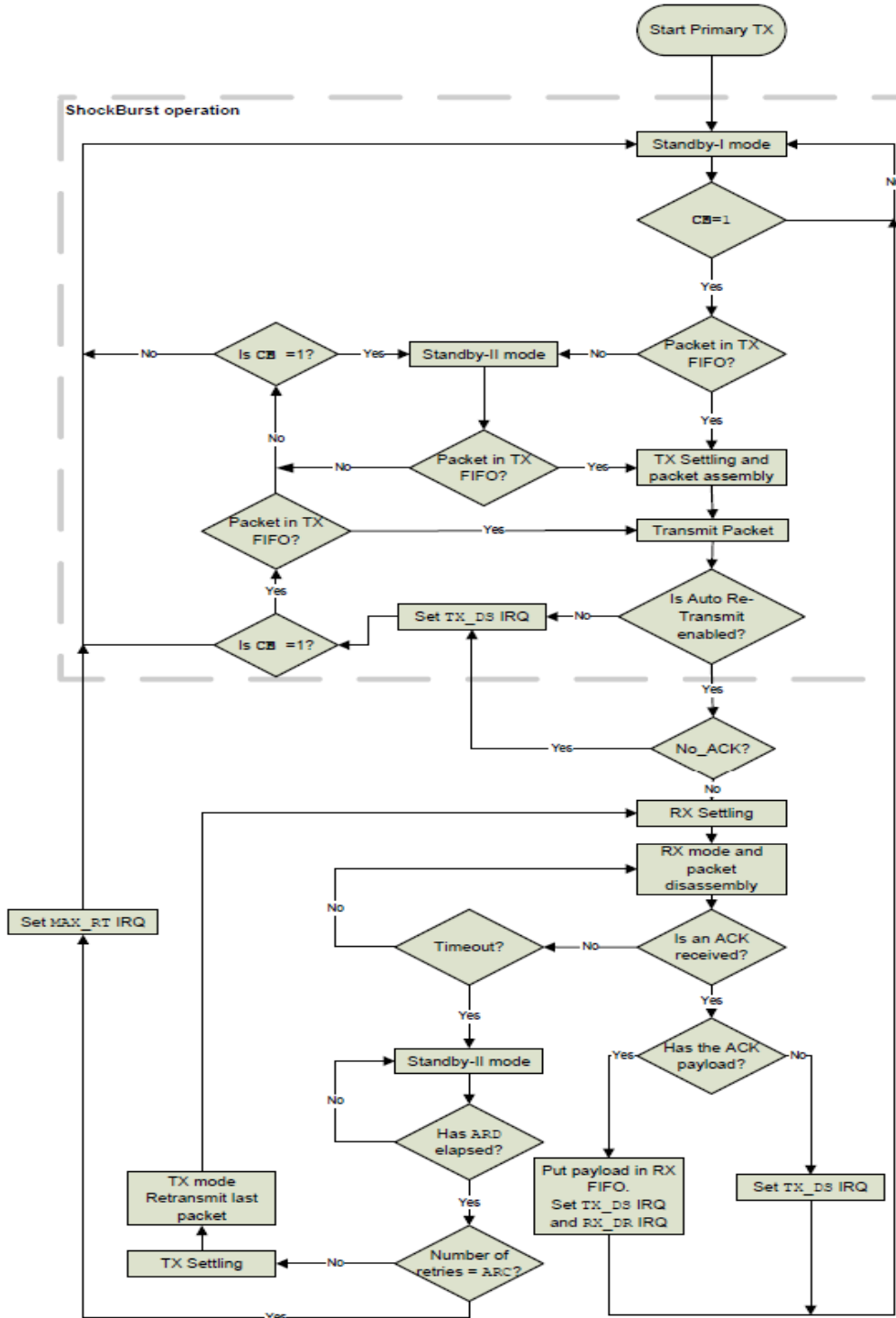
أما PLOS_CNT فيقوم بعد المحاولات الاجمالية لإعادة الارسال. يمكن تصفير هذا المسجل عن طريق الكتابة الى المسجل RF_CH. تستخدم المعلومات من كلا العدادين السابقين لتقييم جودة الاتصال.

كبدل عن اعادة الارسال بشكل اوتوماتيكي يمكن اعادة ارسال نفس المعلومات من خلال التعليمة

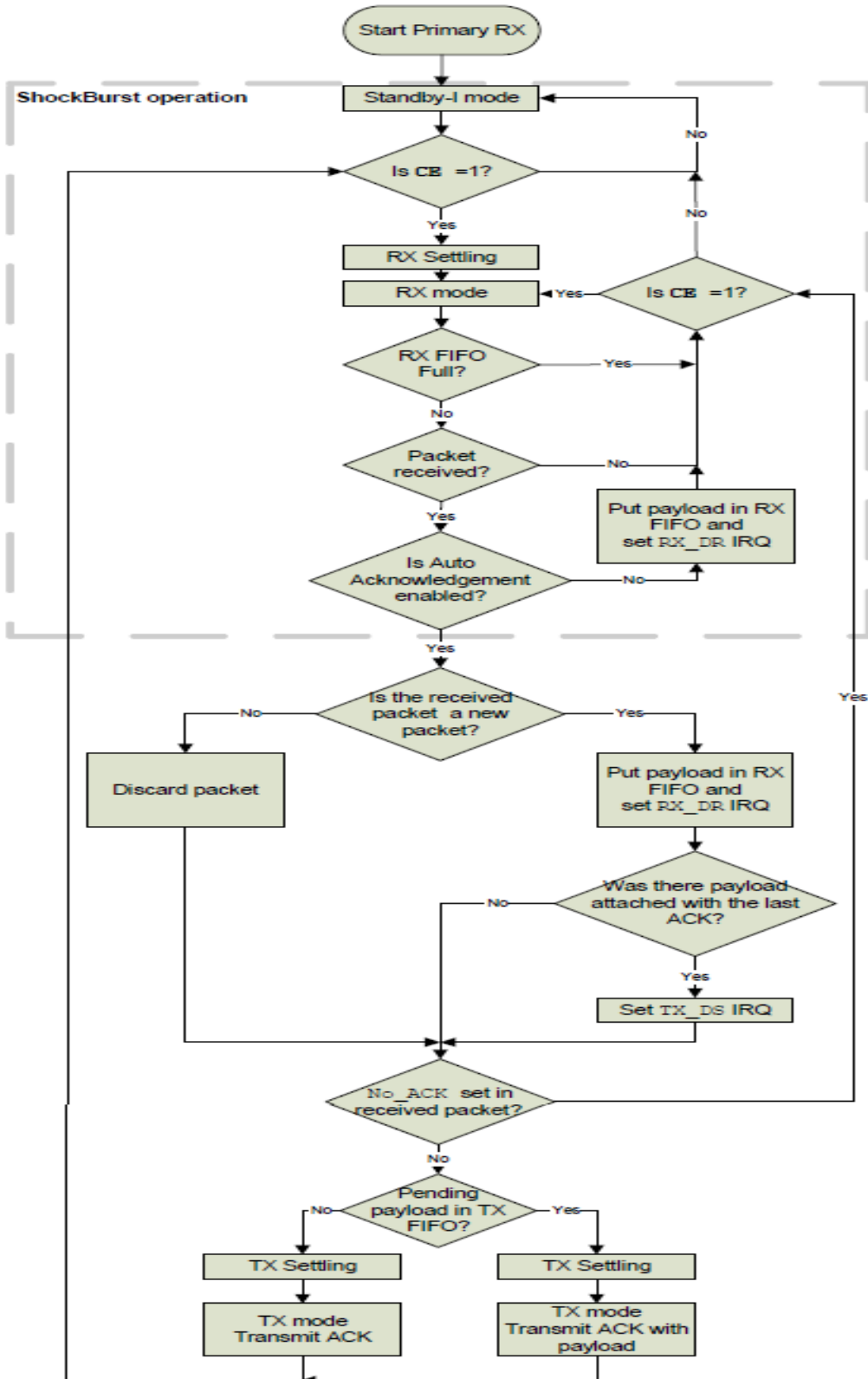
.REUSE_TX_PL

مخططات تدفقية لعمل ESB:

ملاحظة: عمل بروتوكول ShockBurst محدد ضمن الخط المتقطع. عند عمل الجهاز كمرسل (PTX):



رسم توضيحي 21: خوارزمية عمل الـ NRF24L01 كمرسل



رسم توضيحي 22: خوارزمية عمل الـ NRF24L01 كمستقبل

الاستقبال من عدة مرسلات (MultiCeiver):

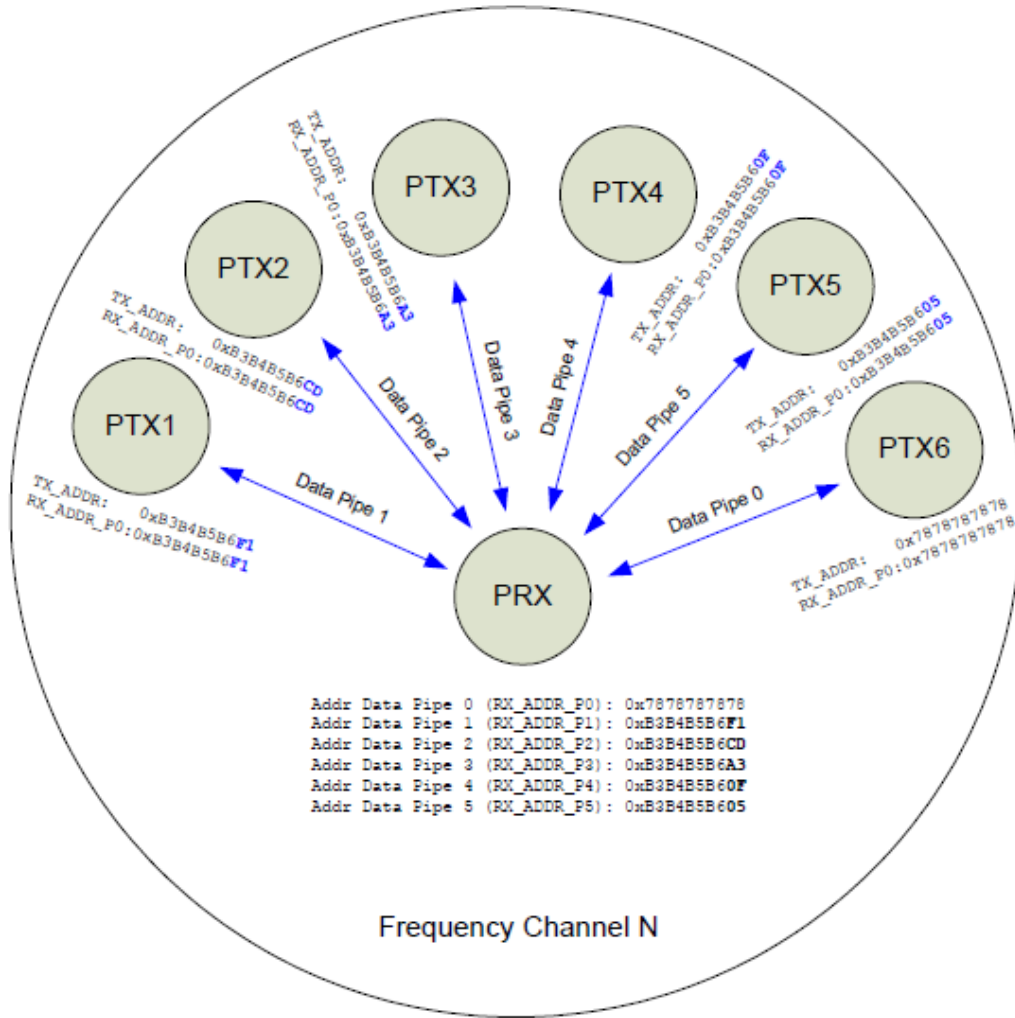
MultiCeiver هي ميزة مستخدمة في وضع الاستقبال RX والتي تمكن الـ NRF24L01 من الاستقبال من 6 خطوط اتصال متوازية لكل منها عنوان Address مميز. جميع هذه الخطوط تتم مراقبتها معا من أجل التأكد من وجود حزم مرسله وعند اكتشاف حزمة مرسله على أحد هذه الخطوط تبدأ عملية الاستقبال. يمكن فقط استقبال حزمة وحيدة على خط واحد في نفس الوقت. تشترك جميع خطوط الاتصال بالإعدادات التالية:

طريقة حساب CRC، طول العنوان المستخدم، القناة الترددية، معدل الارسال وريخ الـ LNA.

يتم تفعيل خطوط الاتصال المرغوبة عن طريق المسجل EN_RXADDR.

يجب الانتباه إلى ان يكون لكل خط اتصال عنوان مختلف عن بقية الخطوط.

خط الاتصال هو عبارة عن قناة منطقية Logical ضمن القناة الفيزيائية. الشكل التالي يوضح هذه الخاصية:



رسم توضيحي 23: الاستقبال من عدة مرسلات في الـ NRF24L01

يمكن ان يتم استخدام عنوان بطول من 3 إلى 5 بايت. الاسلوب التالي متبع عند القيام بعملية العنوان:

خط الاتصال رقم 0 يمتلك عنوان مميز عن البقية. أما خطوط الاتصال من 1 الى 5 فتشترك بالبايتات الأكثر أهمية من العنوان وتختلف عن بعضها فقط في البايت الاقل اهمية. كما هو موضح في الشكل التالي:

	Byte 4	Byte 3	Byte 2	Byte 1	Byte 0
Data pipe 0 (RX_ADDR_P0)	0xE7	0xD3	0xF0	0x35	0x77
Data pipe 1 (RX_ADDR_P1)	0xC2	0xC2	0xC2	0xC2	0xC2
Data pipe 2 (RX_ADDR_P2)	0xC2	0xC2	0xC2	0xC2	0xC3
Data pipe 3 (RX_ADDR_P3)	0xC2	0xC2	0xC2	0xC2	0xC4
Data pipe 4 (RX_ADDR_P4)	0xC2	0xC2	0xC2	0xC2	0xC5
Data pipe 5 (RX_ADDR_P5)	0xC2	0xC2	0xC2	0xC2	0xC6

رسم توضيحي 24: تخصيص العناوين لأنابيب الاتصال المختلفة في الـ NRF24L01

فقط عندما يكمل أحد خطوط الاتصال عملية الاستقبال يمكن لبقية الخطوط أن تبدأ عملية الاستقبال، في حال استخدام عدة مرسلات ومستقبل واحد يجب ضبط ازمدة اعادة الارسل بشكل مختلف لكل من الخطوط بحيث يحجب أحدها البقية لمرة واحدة فقط.

قائمة بالمصطلحات العلمية

A		I	
Acknowledgement	اشعار بالاستلام	Input	دخل
ADC (analog to digital converter)	محول من تشابهي الى رقمي	Internet of things (IOT)	انترنت الأشياء
Address	عنوان	ISM band (Industrial scientific and medical)	الحزمة المخصصة للأغراض الصناعية، العلمية والطبية
Application layer	طبقة التطبيقات		
Assembly	تجميع		
Automatic	تلقائي		
C		L	
Character	محرف	Localhost	المضيف المحلي
Client	عميل		
Control	تحكم		
CRC (Cyclic Redundancy Check)	الفحص الدوري للفائض		
D		M	
Data	بيانات	Master	سيد
Datalink layer	طبقة نقل البيانات	Message count	عدد الرسائل
Destination address	عنوان الوجهة	Multiceiver	مستقبل من عدة مرسلات على عدة انابيب اتصال مختلفة
F		N	
Field	حقل	Network layer	طبقة الشبكة
		Node	عقدة
G		O	
Gateway	بوابة عبور	Output	خرج
H		P	
Hardware in the loop	عتاد صلب ضمن الحلقة	Packet	رزمة
Hop count	عدد قفزات	Payload	حمولة
Hardware node	عقدة مكونة من عتاد صلب	Phase-locked loop	حلقة اقفال طوري
Hybrid node	عقدة هجينة	Physical layer	الطبقة الفيزيائية
		Port	منفذ
		Preamble	سلسلة بتات لتحقيق التزامن عند بدء الارسال
		Proactive	تفاعلي

R

Raw	خام
Receive	استقبال
Routing protocol	بروتوكول توجيه

S

SAR (sequential assignment routing)	التوجيه بالإسناد المتسلسل
Serial	تسلسلي
Server	مخدم
Sink	مصرف
Slave	عبد
Socket	مقبس
SPI (serial peripheral interface)	واجهة الربط التسلسلية مع المحيطيات
String	سلسلة محارف
Synchronization	تزامن

T

Transmit	بث
----------	----

U

UART (Universal Asynchronous Receiver/Transmitter)	معياري عالمي للإرسال/الاستقبال غير المتزامن
UDP (user datagram protocol)	بروتوكول مخطط بيانات المستخدم
USB (Universal Serial Bus)	معياري الاتصال التسلسلي العالمي

W

Websocket protocol	بروتوكول اتصال ثنائي الاتجاه
--------------------	------------------------------

قائمة بالجداول الواردة

7	جدول 1: حقول الرزمة في بروتوكول التوجيه SAR
7	جدول 2: المتحولات الخاصة بتخزين معلومات التوجيه في العقد
12	جدول 3: المتغيرات الخاصة بطبقة الشبكة
13	جدول 4: التوابع الخاصة بطبقة الشبكة
14	جدول 5: التوابع ضمن المكتبة NRF_Tools
15	جدول 6: توابع الطبقة الفيزيائية للبيئة الحقيقية
16	جدول 7: توابع طبقة نقل المعطيات للبيئة الحقيقية
36	جدول 8: تعليمات التحكم بالنرمال NRF24L01 عن طريق الحاسب

قائمة بالأشكال الواردة

- رسم توضيحي 1: مخطط صندوقي لعقدة حقيقية 9
- رسم توضيحي 2: وحدة الهزاز داخل المتحكم PIC12LF1552 10
- رسم توضيحي 3: المكتبات التي يتألف منها كود المتحكم PIC12LF1552 14
- رسم توضيحي 4: مخطط صندوقي لدارة الربط 18
- رسم توضيحي 5: المخطط الصندوقي لل NRF24L01 19
- رسم توضيحي 6: هوائي الدارة المطبوعة 20
- رسم توضيحي 7: دارة الملازمة MAX232 20
- رسم توضيحي 8: مخطط شبكة الحساسات اللاسلكية المدروسة 23
- رسم توضيحي 9: مخطط صندوقي للطريق من عقدة حقيقية الى عقدة المصرف 24
- رسم توضيحي 10: مخطط ارسال البيانات من العقدة الحقيقية الى المصرف 25
- رسم توضيحي 11: وصل SPI سيد واحد وعدة عبيد طريقة الوصل التقليدية باستخدام slave select 30
- رسم توضيحي 12: تبادل البيانات في SPI بين السيد والعبد 31
- رسم توضيحي 13: وحدة SPI ضمن المتحكم 32
- رسم توضيحي 14: ربط متحكمين عبر SPI 34
- رسم توضيحي 15: المخططات الزمنية لـ SPI وفق أوضاع مختلفة 35
- رسم توضيحي 16: شكل رزمة الـ ESB 39
- رسم توضيحي 17: حقل التحكم بالرزمة للـ ESB 39
- رسم توضيحي 18: آلية زيادة الحقل PID في رزمة الـ ESB 42
- رسم توضيحي 19: التركيب التلقائي للرزمة في الـ NRF24L01 43
- رسم توضيحي 20: تفكيك الرزمة في الـ NRF24L01 44
- رسم توضيحي 21: خوارزمية عمل الـ NRF24L01 كمرسل 46
- رسم توضيحي 22: خوارزمية عمل الـ NRF24L01 كمستقبل 47
- رسم توضيحي 23: الاستقبال من عدة مرسلات في الـ NRF24L01 48
- رسم توضيحي 24: تخصيص العناوين لأنايبب الاتصال المختلفة في الـ NRF24L01 49

المراجع العلمية المستخدمة

- [1] Waltenegus Dargie, Fundamentals of wireless sensor networks, Technical University of Dresden, Germany, 2010.
- [2] Falko Dressler, Self-Organization in Sensor and Actor Networks, University of Erlangen, Germany, 2007.
- [3] Microchip. <http://www.microchip.com> .
- [4] Nordic Semiconductor. <http://www.nordicsemi.com>