Titulo:	**Identificación electrónica**

Ciclo Lectivo **2021**	Curso N°	R2051	Grupo N°	**2**

Integrantes	Apellido Nombres	Legajo	Calificación individual	Fecha
Matias Agustin Zapata		R2051		

Calificación grupal:		Fecha:

Profesor:	**Daniel Jose López Amado**
Auxiliar/es Docente:	**Mariano Vedovato**

Observaciones primera entrega	
Observaciones segunda entrega	

Desarrollo de la idea fuerza

Todo proyecto tecnológico surge con el propósito de resolver un problema social, en este caso, el problema que tratamos de resolver es la ineficiencia de los distintos organismos ya sea empresariales o gubernamentales para procesar tramites y validar documentación de distinto tipo.

El tiempo que conllevan dichos procesos suele ser extenso pero su contraparte digital muchas veces carece de la seguridad que es necesaria para ciertas situaciones. Es por ello que creamos un dispositivo que se asocia a una persona en individual y le otorga a esta la oportunidad de firmar documentos digitalmente de manera segura, así como almacenar contactos de interés y generar y almacenar claves seguras.

El proyecto se basa en los pilares de la encriptación asimétrica que está constituida por una clave publica y una privada relacionadas entre sí. Esto permite múltiples aplicaciones, entre ellas, la firma digital que consta de encriptar un mensaje con la clave privada para luego verificar el uso de esta con la clave pública. De esta manera, con una clave privada uno tiene el poder de la firma y por ello es de fundamental importancia que la misma viva en un ambiente seguro, que no se pierda ni sea empleada por quien no debe ser empleada y de allí que surge nuestro dispositivo que pretende brindar la protección necesaria para que se pueda emplear el sistema en cuestión para facilitar los procesos administrativos digitales. Nos proponemos desarrollar una plataforma que incluya las siguientes funcionalidades:

- Encriptación de desencriptación de información para el establecimiento de comunicaciones seguras entre las partes
- Validación de identidad y firma digital de documentos para la realización de todo tipo de trámites
- Generación y almacenamiento de claves seguras
- Almacenamiento en frio de claves simétricas y asimetricas

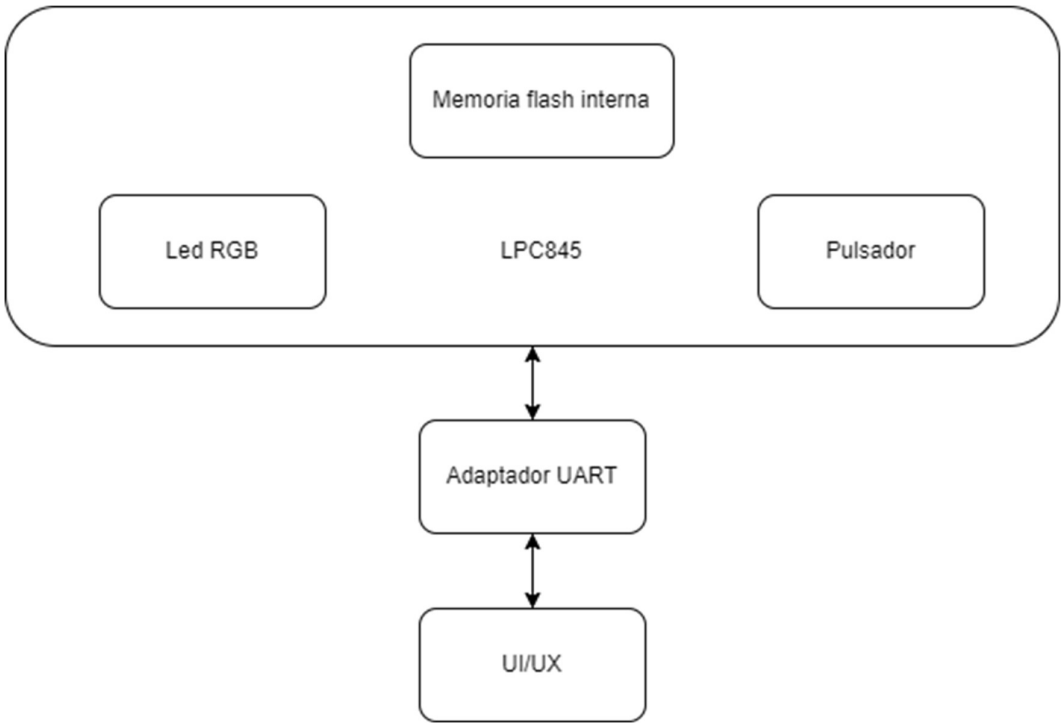
Introducción

Objetivos

Como ya se ha mencionado, se construirá un dispositivo con el objetivo principal de poder habilitar la digitalización de procesos administrativos. Para ello es que haremos uso de la infraestructura de encriptación asimétrica y con el objeto de mantener la clave privada segura (factor fundamental para la viabilidad del proyecto) se trasferirán todas las funcionalidades que de ella derivan, al hardware. De esta manera obtenemos un efecto similar al que se puede observar en las billeteras digitales de almacenamiento en frio, manteniendo el secreto en cuestión de manera offline la mayor parte del tiempo y limitando al máximo el acceso a la mismo.

Proponemos crear un dispositivo capaz de firmar documentos digitales, almacenar y generar claves simétricas y almacenar contactos para evitar problemas de phishing.

Diagramas en bloques



Descripción detallada

Bloques

LPC845

La plaqueta de desarrollo empleada será la encargada de realizar todas las operaciones de interés y la que a través de el adaptador UART comunique los resultados a la interfaz gráfica. El mismo almacenara en la memoria flash interna la clave pública y privada, así como también las claves y contactos que ingrese el usuario a través de la PC. Asimismo, interaccionara con el usuario de manera física a través del pulsador y comunicara los estados de espera y confirmación a través del led RGB presente en la misma.

Adaptador UART

Sera el encargado de habilitar la comunicación entre la PC y el LPC845

UI/UX

La interfaz grafica es la encargada de realizar los distintos pedidos al hardware creado. Entre otras funciones se encarga de calcular el hash de los documentos a firmar, intercambiar claves con el dispositivo, encriptar toda la comunicación, generar claves simétricas random, etc.

Especificaciones

LPC845: El dispositivo ha sido programado con MCUXpresso IDE y hace uso del SDK provisto por el fabricante para el manejo de los distintos periféricos. El mismo esta montado en una plaqueta perforada con el propósito de facilitar las demostraciones.

UI/UX: La interfaz grafica ha sido diseñada en una primera instancia con Figma y se ha implementado en C++ utilizando el framework QT. Para el manejo de la comunicación serial se ha empleado la librería QSerialPort mientras que para las encriptaciones se ha reutilizado los módulos creados para el LPC845. Para el calculo del hash con el algoritmo Md5 se ha empleado la librería QCryptographicHash.

Descripción del Hardware utilizado.

Para el Desarrollo del proyecto se utilizaron dos piezas de hardware, por un lado, la plaqueta de desarrollo LPC845 de NXP y por el otro un adaptador de USB a serial. Respecto a la primera pieza de hardware, se ha hecho uso de la memoria flash, el pulsador, el led RGB y la comunicación serial. La segunda pieza de hardware empleada no posee un software en particular asociado ya que se conecta de manera directa a los pines de transmisión y recepción del primer dispositivo sin mayor configuración de su parte. A continuación, pasamos a explicar los distintos módulos desarrollados para el uso de los periféricos mencionados y las funciones que facilitan las distintas tareas empleadas.

Como ya se mencionó, se ha hecho uso de los siguientes periféricos:

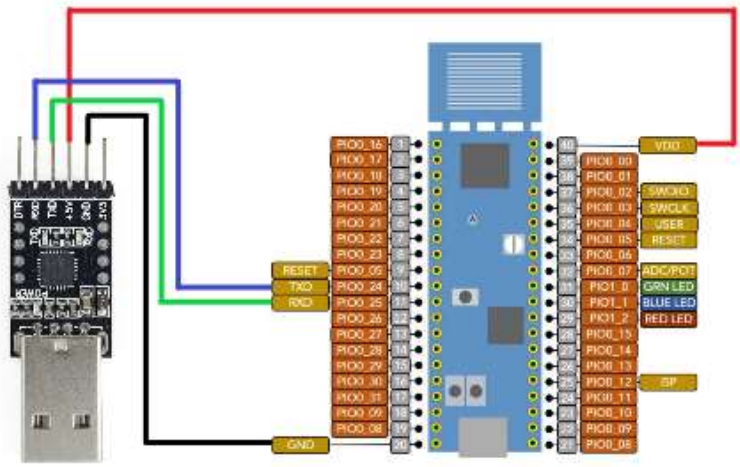
- Memoria Flash: Este periférico ha sido utilizado para almacenar las claves y contactos que ingreso el usuario por la aplicación de QT y el hash producto de la combinación del nombre de usuario y clave de acceso para la autenticación. Para ello desarrollamos dos módulos: Flash y Storage

  El modulo Flash cuenta con funciones wrapper que facilitan ciertas tareas frecuentes en el uso de la memoria flash del dispositivo. Se ha empleado el SDK provisto por el fabricante para el desarrollo de las siguientes funciones:

  - *void Flash_Init(void);* // Inicializacion del modulo
  - *uint8_t Flash_WritePage(uint32_t sector, uint32_t page, uint32_t * buffer, uint32_t size)* // Dado un sector y una página page escribe la data de tamaño size apuntada por buffer
  - *uint32_t Flash_GetPageAddress(uint32_t sector, uint32_t page)* // A partir del numero de sector y pagina se calcula la dirección de memoria y se la devuelve
  - *uint8_t * Flash_GetPageMemCpy(uint32_t sector, uint32_t page)* // Se copia la data apuntada por la dirección correspondiente a sector y page en una memoria dinámica y se devuelve dicha memoria.

  Por otro lado el modulo Storage es el modulo que hace uso del módulo Flash para realizar funciones particulares a la aplicación desarrollada.
- Puerto Serial: Para la comunicación serial se ha hecho uso de la librería provista por la catedra. Se estableció una comunicación a 115200 bps en el puerto serial 0.
- Led RGB
- Pulsador incorporado.

Circuito



Link a hojas de datos

- LPC845
    - https://www.nxp.com/docs/en/data-sheet/LPC84x.pdf
- Conversor Usb A Serial Uart Ttl Chip Cp2102
    - https://www.silabs.com/documents/public/data-sheets/CP2102-9.pdf

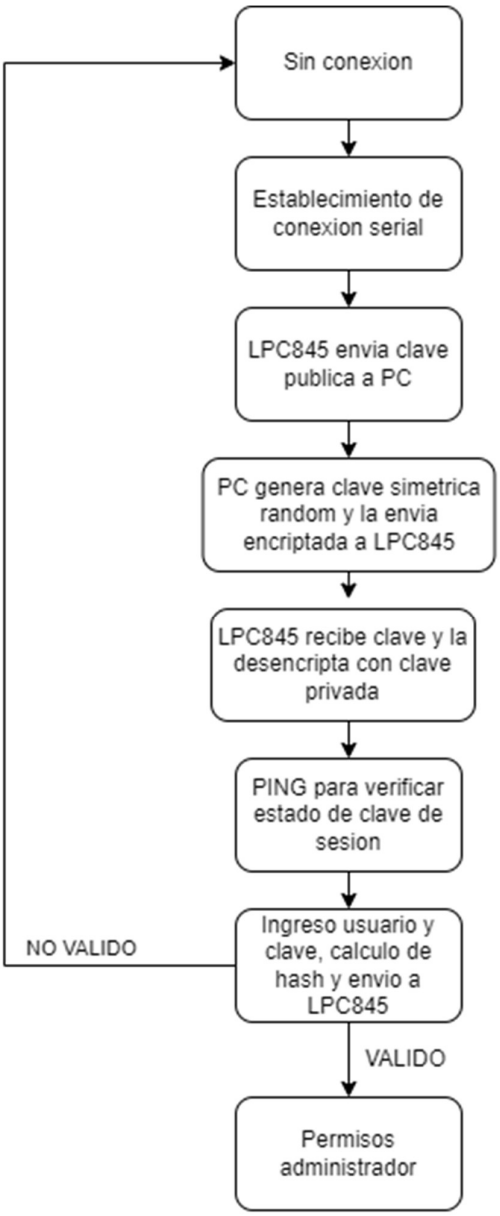
Máquina de estados de la aplicación

El software del LPC845 esta conformado por dos maquinas de estado principales. La primera es llamada TramaMe, la función principal de la misma es recibir y procesar la información recibida desde la PC. La segunda maquina de estados es llamada MainMe y es la encargada de tomar la información recolectada por TramaMe y producir las respuestas a los pedidos de la PC.

Para dejar aún más en claro el funcionamiento de las maquinas de estado empleadas creemos que es importante, en una primera instancia, aclarar cuales son los procesos que son llevados a cabo para cumplir con cada una de las funciones del dispositivo. A continuación, presentamos y explicamos brevemente cuales son estas funciones y los pasos que la conforman para luego si entrar en detalle con el funcionamiento particular de las máquinas de estado.

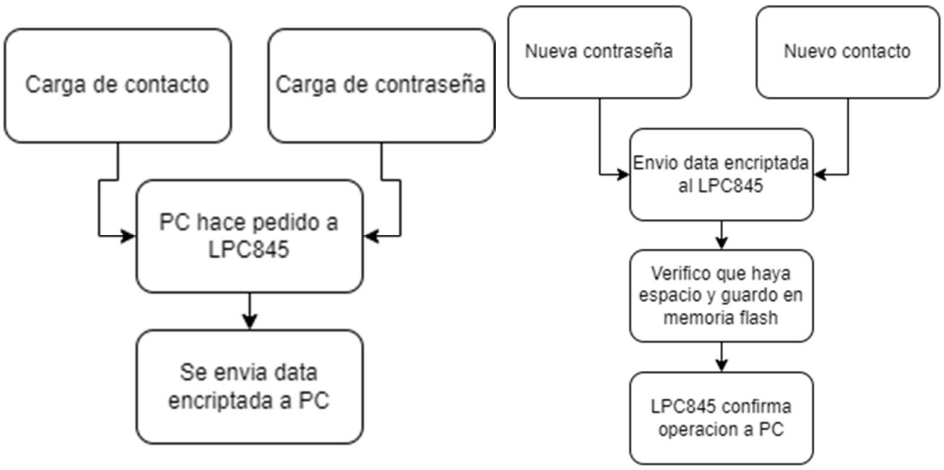
Los principales procesos efectuados en el dispositivo son los siguientes:

- **Conexión y autenticación:** Este es el primer paso en todo flujo de uso. En esta etapa se genera una clave simétrica que será utilizada con el método de encriptación AES128 para encriptar todos los mensajes entre la PC y el LPC845. Esta clave es generada por la PC de manera aleatoria. Luego se pide al LPC845 la clave pública y con esta se encripta dicha clave generada de manera que solo el dispositivo con la clave privada pueda desencriptar el mensaje. Así, la PC envía la clave de la sesión al micro quien la recibe, desencripta y usa para encriptar todo lo que se transmita durante esta sesión. Para verificar que la clave que presentan ambos dispositivos es la misma, se envía un mensaje conocido (“PING”) encriptado, de esta manera la PC sabe que puede continuar la comunicación sin problemas. En esta instancia, la aplicación de PC le requerirá al usuario que ingrese la clave y nombre de usuario que se corresponde con lo almacenado en el LPC845, con la información ingresada, se calcula el hash con el algoritmo Md5 y se envía encriptado por serial. Si el hash coincide con el almacenado en la memoria, se le otorgan permisos de administrador al usuario y se aguarda por mas comandos provenientes de la PC, caso contrario, inmediatamente se cancela todo y se vuelve al estado de sin conexión.



- **<u>Firma de archivos:</u>** La firma de archivos consiste en tomar un archivo, calcular el hash único al mismo con el algoritmo Md5 y luego enviarlo al LPC845 para que este, haciendo uso de la clave privada almacenada en la memoria encripte dicho hash y lo devuelva por serial. De esta manera nos aseguramos de la veracidad de la firma porque poseemos un archivo con un hash único y una string con el hash firmado por la clave privada que es única a la persona en cuestión.

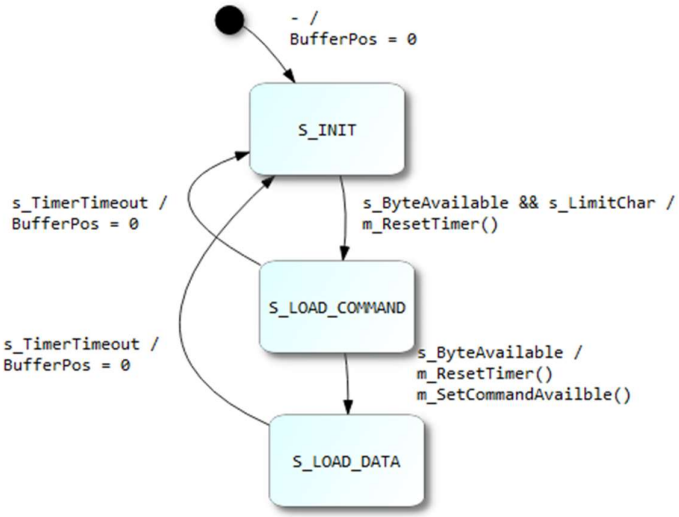
- **<u>Almacenamiento y carga de data:</u>** Si así lo deseamos podemos generar y guardar claves y contactos en la memoria flash del dispositivo creado. Para guardar tanto contactos como contraseñas el procedimiento es el mismo, se requiere el ingreso de la data al usuario, se genera una string de un tamaño fijo que la contenga, se la encripta y se la envía al LPC845 para que la almacene. Cuando lo que queremos hacer es ver los contactos o contraseñas que guardamos lo que debemos hacer es desde la aplicación de QT solicitar la visualización, en ese momento el LPC845 tomara un contacto o contraseña de las almacenadas y las ira enviando de manera encriptada a la PC para que en ella podamos verlas y o eliminarlas.



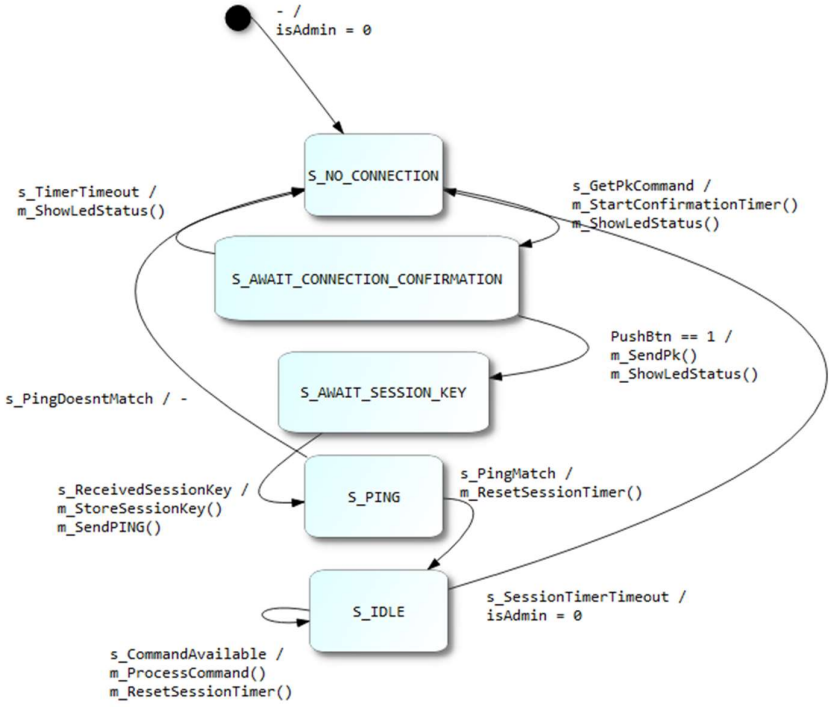
Ahora bien, para poder llevar a cabo todas estas tareas es que, como ya mencionamos, se han desarrollado dos maquinas de estado que serán presentadas a continuación.

TramaMe se encarga de reconocer el inicio de la trama y almacenar el comando y la data recibida. El inicio de la trama se reconoce con el carácter '#', a partir de alli se inicializa un temporizador de 500ms para que en caso de que

no se reciba nada en dicho periodo de tiempo, se de por finalizada la recepción. Una vez recibido el carácter de inicio de trama, se procede a la recepción del comando que nos envía la PC y luego a la recepción de la data. En todos los casos excepto en el caso en que esperamos el carácter de inicio de trama se reinicia el timer cada vez que se recibe nueva información. Para evitar la perdida de data, se envía un byte de confirmación por cada byte recibido.



La segunda maquina de estados empleada lo que hace es tomar el comando y la data capturada por la primera maquina y atravesar el proceso de conexión para después procesar los comandos recibidos.



Problemas encontrados a lo largo del desarrollo del TPO

A continuación, presentamos algunos de los problemas encontrados a lo largo del proyecto.

1. Comunicación serial. En un principio tuvimos problemas respecto a la comunicación serial principalmente por información que no se enviaba o no se recibía correctamente. Se fueron encontrando soluciones particulares y creando convenciones de la trama, pero se logró solucionar.
2. Algoritmos de encriptación. En un principio fue complicado hallar implementaciones de los algoritmos desarrolladas en C que puedan funcionar con los recursos que teníamos disponibles en el micro. Por ese lado logramos encontrar los algoritmos correspondientes y hacerles las modificaciones necesarias para su uso. Así mismo las mismas hacen un uso intensivo de memoria dinámica, lo cual, si bien no llego a ser un problema importante, si una molestia que muchas veces retrasaba el desarrollo del proyecto.
3. Modificación de los objetivos y manejo del tiempo. La modificación en el numero de integrantes del grupo a lo largo del desarrollo del proyecto fue un problema por la redefinición en los objetivos del proyecto que forzaron el descarte de mucho trabajo ya hecho y produjeron la reprogramación de las tareas.

Beneficios encontrados a lo largo del desarrollo del TPO

Particularmente creemos que el desarrollo del TPO ha sido beneficial para nuestro aprendizaje. Por un lado es bueno tener la motivación de aplicar los conocimientos adquiridos a lo largo del año en un proyecto de interés personal. Por otro lado, creo que es muy beneficial el hecho de encarar un proyecto de magnitud, no solo para ir tomando perspectiva de los tiempos y problemas que pueden ir surgiendo, si no también para uno ir ganando experiencia y confianza en lo que se puede hacer.

Desde su lugar de alumno que elementos agregaría o quitaría para el desarrollo del TPO

Desde mi punto de vista yo creo que mejoraría un poco la planificación. Creo que quizás sería más útil en vez de hacer reportes semanales de lo hecho, en un principio realizar una buena descripción de lo que se va a desarrollar, de las tareas que hay que realizar y establecer checkpoints fijos. De esta manera se podría lograr un mayor compromiso por parte de nosotros los estudiantes y se tendría más información del estado del proyecto, lo faltante y lo completo. Siempre sujeto a posibles replanificaciones.

Conclusion

Como conclusión podemos decir que si bien, respecto a la idea inicialmente planteada, hubo cambios en las metas del proyecto, se pudo cumplir con una versión mínima viable de un proyecto que tiene la capacidad de expandirse. Creemos que se ha respetado el espíritu del proyecto y que la académicamente ha sido de gran ayuda para fijar los conceptos estudiados y prepararnos para lidiar con todo aquello que atraviesa la frontera de lo conocido.

En un futuro dicho proyecto se podría expandir para adquirir una mayor practicidad con funcionalidades como almacenamiento de archivos, conectividad para enviar y recibir información de manera automática, formularios predeterminados, distintos tipos de ingresos y permisos, etc.

Bibliografía

- Implementacion de AES en C - tiny-AES-c: Small portable AES128/192/256 in C (github.com)
- AES128 - Advanced Encryption Standard
- Algoritmo de reducción criptografico MD5 - MD5
- Sistema criptografico de clave publica (Rivest, Shamir y Adleman) - RSA
- Documentacion Qt - Qt Documentation
- Datasheet LPC845 - LPC84x.pdf
- Datasheet CP2102-9 - CP2102-9.pdf