

SMS Inbox project

Overview

Our client - - must be able to receive SMS from an allowed list of senders ONLY.

Our project our project only has to be concerned with the receiving of the SMS and directing the content of the message along with the metadata of the message to the serial port of the server that the device which received the SMS is connected to.

I want to use an Arduino GSM shield to receive the SMS on a SIM card. We must output the SMS to the serial port via a USB cable s **Phase one** of the project. **Phase two** of the project will send the output of the SMS to the local server's MySQL instance. The Linux server that is connected to the USB cable must have a listener program watching for data arriving on the serial port. **The ultimate aim is to save the received SMS into the server's MySQL database** and have the able to be queried by our client using a C#/C++ program we have to write for him to place on his Windows server(s).

This project is therefore in two parts

- Part 1 is to build the Arduino GSM shield and output SMS messages to the serial port.
- Part 2 is to build a program in C# or C++ that will reside on the client's Windows server to listen for incoming SMS messages.

This Arduino GSM shield will be connected to our own Linux server and will not be placed anywhere near the client infrastructure. We should be able to cluster to these Arduino GSM Shields and expand our capacity for receiving SMS as and when required.

Our Linux server will connect to our client's environment via a VPN secure tunnel or mpls connection.

To plan our development environment all we need is one Arduino GSM shield connected to a Linux computer which can be a desktop or a laptop that is connected to the internet.

Our *live development testing environment* is rack-hosted Linux server running Ubuntu or centos.

We will also have a **production environment is a pair of mirrored Linux servers** one sitting in 365MD's office (our distributor in South Africa) and the other sitting in Data Keepers data centre in Cape Town on the Seacom cable.

Project instructions:

task	Description	Done?
	Part one	
1	Install the Arduino GSM shield	
2	Connect GSM shield to Linux server with USB cable	
3	install SMS source code from sample script to receive SMS	
4	Output one received SMS to serial port	
5	Catch the “outputted” SMS on the serial port on the Linux server	
	Part two	
6	Write a INBOX C++/C# program that will capture the serial port data and list each message and save this to a MySQL database on our SMS server.	
7	Write a client side C# or C++ program to query our MySQL database to check for any unread messages collected from the serial port output. Include in the c program in task 6 the ability to delete the message from the SIM card as soon as it has been read	
8	Create a log file on our Linux server that will record the message metadata which is: 1) From MSISDN 2) To MSISDN 3) Timestamp received (HHMMSSDDMMYYYY) 4) Message body(content) 5) any other relevant data from the meta data This log must be encrypted.	
9	Make it possible for the client to query the log file by requesting any message ID or a range of time with a from and to limit, in other words search for all messages between 10 a.m. and 11 a.m. or search for messages with IDS from this number two that number, or just search for a single message ID. The report should provide the following information: 1) Message ID requested 2) Message status (found, not not found) 3) Timestamp received	

	<p>4) Any message metadata</p> <p>This query must be via HTTP. For example, client will send a HTTP POST/GET to our SMS-receiving-server that has our Arduino GSM shield connected to it. The server must respond as above to the client query.</p>	
--	---	--